ADST/WDL/TR--95-W003337C

Advanced Distributed
Simulation Technology

Functional Description Document for ModSAF

Date: 28 April 1995
ModSAF Version 1.51

Prepared for:

STRICOM
U.S. Army Simulation, Training and Instrumentation Command
12350 Research Parkway
Orlando, FL 32826-3275

Contract N61339-91-D-001
Delivery Order ModSAF
Delivery Order 0058

LORAL
ADST Program Office
12151-A Research Parkway
Orlando, FL 32826

REPORT DOCUMENTATION PAGE

Form approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		ModSAF Version 1.51

4. TITLE AND SUBTITLE
Functional Description Document for ModSAF

5. FUNDING NUMBERS
Contract No. N61339-91-D-0001
Delivery Order ModSAF
Delivery Order 0058

6. AUTHOR(S)
Loral Advanced Distributed Simulation
50 Moulton Street
Cambridge, MA 02138

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Loral System Company
ADST Program Office
12151 -A Research Parkway
Orlando, FL 32826

8. PERFORMING ORGANIZATION REPORT NUMBER
ADST/WDL/TR-95-W003337C

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S Army Simulation, Training and Instrumentation Command (STRICOM)
Naval Air Warfare Center, Training Systems Division
12350 Research Parkway
Orlando, FL 32826-3275

10. SPONSORING ORGANIZATION REPORT NUMBER
A006

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)

The Functional Description Document for ModSAF 1.51 provides a functional overview of the ModSAF system, including a description of the system components, entity simulation, and unit simulation.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	17. SECURITY CLASSIFICATION OF THIS PAGE	17. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Functional Description Document for ModSAF

V1.5.1

28 April 1995

LADS Document Number 95007 v1.0

Table of Contents

1 Introduction

This introduction includes some of the newly added and modified functionality of ModSAF V1.5.1:

1.1 ModSAF V1.5.1 Enhancements

- RWA RF Hellfire Libvhellfire implements a vehicle-level task that handles autonomous Hellfire missile shooting. That is, laser designation and Hellfire missile shooting is done by the same vehicle.
- RWA Damage The Damage vulnerability tables, as given from AMSSA, are different for RWA vehicles. Instead of having a catastrophic, mobility, firepower, and mobility/firepower kill RWA vehicles have Attrition, at least forced landing, and at least mission abort.
- Air Defense For certain Air defense teams (US M2 Stinger, US DI Stinger, USSR SA 15, and USSR 2S6 teams), the method for which the did flyout had to be changed from dynamic to probabilistic. The parameter reader files, for these teams, had to be change to incorporate the Ballistic gun tasks to handle the fly outs for there missiles.
- New Vehicles/Units

```
US NLOS
US GBS-FAADS
US M102
US M198
US XM8
US M35A2 FDC

USSR 2S12
USSR 2B11

US M109 BATTERY
US M109A1 BATTERY
US M109A3 BATTERY
US M109A5 BATTERY
US M109A6 BATTERY
US M1064 PLATOON
US M106A1 PLATOON
US XM8 PLATOON
US M102 BATTERY
US M198 BATTERY

USSR 2S12 BATTERY
USSR 2B11 BATTERY
USSR 2S19 BATTERY
```

```
USSR 2S1 BATTERY
USSR BM21 BATTERY
```

- Simulation of Smart Munitions

```
US SADARM
US NLOS

USSR 120mm MCS SMART
USSR 152mm MCS SMART
USSR 120mm MMB SMART
```

- Minefields – Withdrawal from minefields was enhanced. Minefield breaching, using full-width mine plow was added. The Grizzly engineering vehicle was added with minefield breaching capabilities.
- AVLB – Added an AVLB vehicle that can move to a breach position and deploy its bridge over the obstacle. Other vehicles can then drive over the deployed bridge. Any AVLB vehicle without a stowed bridge can retrieve a laid AVLB bridge. AVLB entity articulated parts are positioned realistically.
- RWA Autonomous Hellfire – Implements a self-designating, laser-based Hellfire. It uses the DIS 2.0.3 LASER PDU. Two new tasks were added to implement RWA Autonomous Hellfire.
- RWA Remote Ground Laser Designation – Implements a first cut at having a ground-vehicle remotely designate for another vehicle. It uses the DIS2.0.3 LASER PDU.
- Mobility and Firepower Damage/RWA Mission Aborts and Forced Landings – Three tasks written to deal with firepower RWA damage: a mobility kill in which the RWA is forced to land, firepower kill in which the mission is aborted, and mobility kill in which the RWA stops the mission abort and spawns a forced landing.
- Longbow Radar – Libpdetection provides a place to store the probability of detection models used by '`Genradar`'.
- Nap-of-Earth fly route – Improved Nap-of-Earth flying mode during a '`RWA flyrte`' task. The flying route passes concealed areas (from enemy) as much as possible.
- Visual_scanners can be restricted to inside or outside the field-of-regard (FOR).
- When a vehicle is selected on the PVD, that vehicle's field-of-regard (FOR) is graphically displayed. The radius of the drawn pie-slice is the range of the gunsight.
- After a vehicle is acquired by a sensor, a time delay is imposed before that vehicle is eligible for assessment for target selection. This time delay is specified for each sensor.
- Field-of-View Sensor field-of-view (FOV) revised in '`visual_macros.rdr`' to incorporate data from AMSAA. Target acquisition may appear to occur more slowly due to the significant reduction in FOV.
- '`Visual`' modified so that it performs a separate intervisibility call from each sensor's perspective. For example, driver-sight is ineffective if blocked by a berm.

- Gunsight attachment points changed from main gun to primary turret, with the same offsets used for other turret sights allowing visual to use the same intervisibility call for all turret sights.
- Vehicle dimensions revised in '`physdb.rdr`' to incorporate data from AMSAA. Macros were added for optical, ii, and ir contrasts for Knox and SWA terrains.
- VSpotter's management of sensor and spotter data rewritten to accomodate the target acquisition-to-engagement time delay.
- Battalion March – Implements basic battalion-level travel.
- New DI Hull model – Improved DI Hull model, introduced fatigue model, and DI postures (standing, kneeling, prone).
- Company Actions on Contact – A company-level reactive task that monitors enemy activity and reacts to contact.
- New Radio Architecture – Genradio is now a service that handles the incoming signal/radio PDUs.
- UAV Spot report task - Supplies spot reports of enemy locations that can be used by other entities for targeting.
- Inverted SAFOBJ Architecture – This library is a service which allows the creation, deletion, and ticking of local and remote entities.
- '`Libentity`' handles more than two articulated parts.
- Status messages show fully in the message log.
- Multi-resolution Wheeled Vehicle Dynamics – Libwheeled library computes tick dynamics of wheeled vehicles. Dynamics and simplistic driver functions are computed. Also, a resolution level is computed although it is currently used in a token way only. Dynamics runs underneath network levels.
- Unit-level Change Formation Task – Changes the formation of ground level vehicles to that which is specified by the user.
- Mixed-level Change Formation Task – Allows mixed-level platoons to change formation.
- Smoke puff support added
- Editor to allow environmental parameters to be displayed and/or set added
- Support for data-table driven extinction coefficients added
- Flare, flare/smoke signal detection, and task '`Transition on Signal Detect`' added.
- Choice of final state in company attack and unit assault tasks. A new editor appears on these task editors. This new editor toggles between "Secure Objective" or 'Not Secure Objective," with "Secure Objective" as the default value.
- Manned simulator activation, deactivation, and reconstitution from a ModSAF station.
- Real bridge remote demolition
- Unit control measures specified

- Trigger line in Occupy Position

Refer to the *ModSAF Version Description Document*, V1.5.1 (VDD) for information regarding these functions. Capability tests are located in the *Acceptance Test Procedure Manual*, Volume 3.

1.2 Product Overview

`ModSAF` (Modular Semi-Automated Forces) is a set of software modules and applications used to construct Distributed Interactive Simulation (DIS) and Computer Generated Forces (CGF) applications.ModSAF modules and applications let a single operator create and control large numbers of entities that are used for realistic training, test, and evaluation on the virtual battlefield. ModSAF contains entities that are sufficiently realistic resulting in the user not being aware that the displayed vehicles are being maneuvered by computers, rather than human crews. These entities, which include ground and air vehicles, dismounted infantry (DI), missiles, and dynamic structures, can interact with each other and with manned individual entity simulators to support training, combat development experiments, and test or evaluation studies.

ModSAF uses the concept of "selective fidelity" to balance cost, desired performance, and realistic simulation. This concept emphasizes efficiency and the avoidance of the simulation of behaviors and mechanisms that do not produce significant externally visible signatures. For this reason, many models include elements of human control that effectively simplify the behavior of the entities.The ModSAF system architecture consists of C libraries. The libraries have strictly defined interfaces and are layered so that they depend on lower-level libraries only. You can extend the system by replacing or adding new libraries. You can also use ModSAF libraries as components in your own system.Further extensibility is provided through the use of the parameter database. ModSAF parameterizes both behavioral and physical models so that a variety of systems are represented.

At program startup time, the offline parameter database which defines these parameters is translated into a runtime database. This database of parameter files allows modification of entity configuration and behaviors, without requiring the modification of any software.

1.3 Virtual Database

ModSAF components share simulation and control information by using the following virtual databases:

- Simulation Database - Contains information about the physical state of the battlefield and its entities. This information includes entity state as well as impact, collision, and fire events.
- Persistent Object (PO) Database - Transfers initialization, command and control, and system parameter information. This database includes missions and their states, unit hierarchies, simulated communications, and physical model parameters.
- Parameter Database (physical and tactical parameters) - Enables ModSAF components to share entity, unit, and mission component data.
- Terrain Database - Provides information about the battlefield terrain.

Note: These databases are located on the computer, you can access them through the network using NFS protocol.

The following figure shows a SAFstation and a SAFsim communicating by network with PO (Persistent Object) and Simulation packets. When you create objects on the SAFstation, PO packets whose data represents the state of each object are projected onto the network. The objects they represent can then be simulated by SAFsim. When a simulated object moves or fires, SAFsim sends its simulation packets to the network.

```
                           (Terrain Database)
                           (Parameter Database)
                     User             |
                        |                  |
               + ---- | ------- +     |    + -------- +
               |   SAFstation   |<-- + -->|  SAFsim   |
               |                |          |           |
               + -------------- +          + -------- +
                      |                         |
                      |                         |
   Simulation Packets --> ===== NETWORK= ====== <-- PO Packets
```

ModSAF components require only one network to communicate both simulation and control information. Ethernet is supported as a communications medium.

ModSAF can run as one of the following::

- Logger - Use to record activity.
- SAFstation only - Use when heavily monitoring the battlefield, performing terrain analysis, or issuing numerous orders.
- SAFsim only - Use to get the maximum number of vehicles simulated.
- Combined SAFstation/SAFsim - Use to simulate a small number of vehicles and not heavily use the graphical user interface (GUI) (avoids the user interface taking computational resources from the simulation). Running a combined system (also known as a pocket system) is useful for code development or experiments involving small numbers of entities.

2 ModSAF Components

This chapter contains an overview of ModSAF capabilities and organization, network configuration, ModSAF exercise configuration, and logger configuration.

The ModSAF software is organized as follows:

- The SAFstation is a ModSAF workstation which contains the graphical user interface (GUI) and plan view display (PVD).
- The SAFsim program runs on a workstation when one or more workstations provide SAFstation capabilities.
- The ModSAF program provides SAFstation and SAFsim capabilities combined in one executable program, and has the capability to run on a single computer.
- The Logger program runs keeps records of activities.
- The Blaster program runs independently on an isolated network for testing purposes only. It evaluates ModSAF's ability to handle network traffic and also generates randomly placed remote vehicles.

Note: A single computer can run ModSAF, SAFsim, Logger, and Blaster programs but cannot run two simultaneously.

2.1 Network Configuration

The network configuration determines how individual ModSAF systems interact with each other. Network configuration is accomplished by the simulation exercise ID and by the PO database ID.

Only computers using the same simulation exercise ID can interact. If two computers are on different simulation exercises, one cannot interact with the simulated objects of the other. This feature lets different training exercises run on the same physical network since each exercise ID defines a different battlefield.When two or more systems are on the same exercise ID, they can interact with the other's simulated entities using appearance, fire, and detonation packets. Each interacting computer uses the simulation packets to build a shared simulation database that describes the state of the physical battlefield.

In addition, ModSAF uses PO packets to share command, control, and mission state between computers. These packets support the PO database used by ModSAF to request and control the simulation of ModSAF entities. PO database IDs are also used to create independent PO databases.

Note: Only ModSAF computers using the same exercise ID and the same database ID can share command and control information. Having the same database ID is not sufficient since having the same command and control information for two different battles is not recommended.

2.2 ModSAF Exercise Configuration

You can configure a ModSAF exercise in a variety of ways. The precise configuration you use depends on hardware costs, available workforce, and exercise or experiment goals.

Increasing ModSAF Entities

To increase the number of ModSAF entities that a SAFstation controls, more SAFsim computers with the same exercise ID and database ID can be run. When the SAFstation requests that vehicles be simulated, the SAFsim systems determine, based on current loading, which SAFstation should do the simulation.

Placing multiple SAFstations on the same PO database enables the SAFstations to identify each other's overlays and control each other's vehicles. This allows two operators to work simultaneously resulting in more flexible control of ModSAF forces. For example, if the forces controlled by one operator are under attack while the other operator's forces are not, the free operator can control the other operator's force resulting in more realistic behavior.

Summary:

SAFsim processes working together provide a simulation server that responds to commands from users at SAFstations.

When exercises that require detailed or exacting human control of ModSAF entities, or human operator interaction with simulators using radio exist, additional computers can be allocated for SAFstation use. Multiple users at multiple SAFstations can control entities simulated on one or many SAFsims.

2.3 Logger Configuration

If the Logger resides on the same PO database, it can acquire information regarding unit hierarchies and the state of vehicle and unit missions on the database. This feature permits the creation

of ModSAF scenario files from a Logger file. With the scenario files, ModSAF can restart entities from SAFsims on that PO database at any point on the tape.

Note: If only the physical state of the battlefield is necessary, you can turn off the PO packet logging to record only the simulation traffic for that exercise ID.

3 SAFstation

SAFstation interacts with SAFsim to produce a credible force. You can set up, view, control, and participate in a simulation exercise. The SAFstation has a color map display showing the current state of the battle as known by ModSAF. In addition, it provides support for transmitting orders to, and receiving information, from the simulated force.

SAFstation hardware consists of a monitor, a keyboard, and a three-button mouse. A graphical user interface (GUI) lets you access an extensive list of commands and options for monitoring and controlling ModSAF vehicles and assets.

SAFstations can access the simulation and PO databases. The simulation database determines the locations, velocities, and physical conditions of entities. The PO database shares command and control as well as system information with the SAFstations, SAFsims, and Loggers.

A ModSAF commander issues orders to the SAFstation appropriate to the level of command. The system automatically interprets those orders and generates unit and vehicle behavior and tactics without further action from the commander.

Default values are provided by ModSAF and the GUI. Choice-dependent defaults are available on the editors. For example, if you do not specify altitude, and then choose contour flight instead of nap-of-earth, the default altitude changes to a more appropriate value. However, the commander can modify, override, or interrupt any automated system behavior as long as battlefield physics are not violated (for example, vehicles cannot move faster than physically possible). The commander can temporarily replace and resume a task in the execution matrix.Once a mission has begun execution, its status is monitored using the map, the message log (which simulates the receipt of radio reports), and the status display. The ModSAF commander can view and edit the parameters of a mission at any time. This includes editing a route way-point, the engagement criteria, and the reaction to an attack.

3.1 User Preferences

You can set aspects of the user interface according to your preferences. For example, you can set numerical units, coordinate systems, select editing mode for line graphics and editor window display, set time format, zoom scales, and scroll bar display.

You can save these user settings to a file and later retrieve and edit them. You can also override the settings on a case-by-case basis.

In addition, you can select units for the following:

- Distance – Used for distance displays, messages, and user input. The available units are feet, meters, nautical miles, miles, and kilometers.
- Altitude – Used for altitude displays, messages, and user input. The available units are feet, meters, miles, K feet, and kilometers.
- Speed – Used for speed displays, messages, and user input. Available units include knots, miles per hour, meters per second, mach, feet per second, and kilometers per hour.
- Angles – Used for angle displays, messages, and user input. The available units include degrees, mils, and compass direction.
- Fuel – Used for fuel displays, messages, and user input. The available units include liters, gallons, and pounds.

3.2 Map Adjustments

SAFstation provides a two-dimensional view of the simulated environment. This two-dimensional view is a plan view display (PVD) or a tactical map. The SAFstation lets you access various tools for controlling the view onto the environment and determining which features to display.

Terrain Features Display

In addition, the SAFstation lets you determine which terrain features to display on the tactical map. The terrain features available for display include hyposometric background; trees, tree lines, and tree canopies; political boundaries; contour lines; water and soils; buildings, power pylons, and structures; railroads; towns; roads; pipelines; powerlines; and grid lines. You can also choose the hyposometry method to be colored or dithered. You can adjust the update rate, map notations, vehicle picture scale, and icon size as well.Using the SAFstation, you can zoom and pan (select any portion of the map for display) to any point on the PVD map.

You can zoom in on a small portion of the tactical map or zoom out to a larger portion by clicking the mouse button once while the pointer is on the map. In addition, you can select a rectangular area to zoom in on.

You can pan by using scroll bars, dragging a viewing window with the mouse, or placing the pointer on a new map center and clicking the mouse button.Grid lines are shown at a scale appropriate for the current map scale. For instance, at the 1:200,000 map scale, one-digit UTM grids are shown, while at the 1:50,000 map scale, two-digit UTM grids are shown.

A scale menu lets you change the scale of the map display. This menu displays the current scale and provides the option to select a different scale. You can choose whether the map scale is restricted to standard map scales or can be freely adjusted.

3.3 Terrain Analysis

The SAFstation lets you display elevation by hyposometric tinting (the display of elevation by colors) and by contour lines. A parameter file is available so that you can edit the setting of the contour interval.

You can select a point on the map and query the terrain database for information. This information includes:

- Location (in UTM, longitude/latitude, or Cartesian coordinates)
- Soil type (RCI250, road, water, etc.)
- Elevation
- Maximum gradient

The SAFstation has a terrain tool for measurement and intervisibility calculations. Using this tool, you can:

- Measure distances on the map in a user-specified measurement unit.
- Display the difference in elevation between two terrain points selected from the map in a cross-section display. The length and direction of the cross-section line drawn between the specified points are displayed; the composition of the terrain is not.
- Show intervisibility between individual entities, intervisibility between points on the terrain, and area intervisibility around a location or around a vehicle. You can set the range for area intervisibility plots.

You can specify the height above the terrain by which the intervisibility is measured. For air vehicles, the altitude of the vehicle is used as the default height by which the intervisibility is measured. For ground vehicles, the elevation of the commander or driver is used as the height by

which the intervisibility is measured. When intervisibility lines between entities are displayed, the percentage of an entity that is visible is indicated by the color of the intervisibility line and by a text field showing the percentage.

3.4 Mission Graphics

SAFstation provides graphical symbols for mission planning. These graphics are created using a point editor, a line editor, an area editor, or a text editor. You can access these editors by selecting an editor button on the user interface. Once the selected editor appears, set the graphic parameters from the editor menu and place the graphic on the terrain by clicking the mouse button on the color map.

You can save an arrangement of graphics in an overlay file for recall and use in an exercise, or you can modify and save the arrangement as a new overlay. Overlays can be shared between SAFstations, although controls are placed based on which overlays workstations are representing in an exercise (limits access to opposition overlays). By providing the capability to show or hide specific overlays, the SAFstation lets you superimpose graphics on the tactical display.

You can create and edit the following graphical control measures:

- Points – General, coordinating, contact, control, target reference, fortification, decision, hide, and launch points are supported.
- Lines – Plain, front, minefield, fortification, berm, antitank ditch, and wire lines are supported. Lines define routes for entities to follow during a mission (including circular routes). When a route contains road segments the system automatically determines the shortest sequence of road segments between user-specified road points. Lines can also define military control measures, such as unit boundaries, objectives, phase lines, assembly areas, and battle positions. Some line types are available only for map annotation.
- Areas – Areas are closed lines placed on the display.
- Text – Multiple lines of text placed on the display.

Use these graphics in the mission specifications for units and entities. Various colors are provided for each graphic; line and point graphics display as solid or dashed lines.

Select the editing mode used for graphics by choosing either whole or part editing. You can add points to and delete points from a multi-point graphic or delete entirely. Labels can be edited and/or added to graphics.

3.5 Entities and Units

Units, from the company level to the individual entity level, can be created on the SAFstation through a *unit editor* facility. The unit editor lets you indicate the type of unit to be created by selecting the appropriate element of a menu. It also lets you specify that unit's placement on the terrain by clicking on the mouse button while the pointer is on the colormap. Unit control can be transferred between workstations.

You can designate (or label or call sign) units and entities and specify whether to display these designations. If you do not specify a designation for a unit, ModSAF attempts to supply a default designation.

Command control is the highest level at company level. However, it is possible to control at lower levels of command (such as platoon) downward to entity level.

3.6 Situation Display

The Plan View Display (PVD) shows the current situation in an exercise by displaying the current positions of all entities involved in the exercise. The refresh rate of the entity icons can be set from 1 to 120 seconds. You can also temporarily freeze entity icon updates.

Displaying Military Units

Army symbology, defined by "FM 101-5: Operational Terms and Graphics," displays military units on the map.

You can aggregate the display to show only higher-level units or deaggregate the display to show individual entities. The individual entities can be displayed by the standard military symbology or in a non-military picture form. The standard military form shows entity direction quantitized to eight (8) directions (45 degrees) while the non-military picture form shows exact entity and turret orientations. Both forms show catastrophic entity damage drawn in black. The SAFstation has the ability to increase and decrease the drawing size of military or non-military symbols.

Querying Entity Icons

It is possible to query entity icons or pictures to get a description of the entities they represent.

This description includes the entity ID, designation, location and, if the entity is being controlled by the SAFstation making the query, the current speed of that entity.

Displaying Simulation Events

The SAFstation displays simulation events such as indirect fire and collisions. It also displays direct fire events, designating the target and firer, and whether the shot was a hit or a miss. Artillery impacts are displayed as dashed points labeled *Shell*. After two minutes without further artillery, the point disappears.

In addition, the SAFstation can display a continuous update of the mission status of any entity you specify.

Gathering Opponent Information

A ModSAF commander learns the situation, strength, and disposition of the opponent by the units that are in contact with the enemy. The amount of information the units gather is limited to what they can infer from the visible (or electronic) information available to them. Selecting a unit displays that unit's perception of enemy forces. Ground units are displayed as generic boxes, while FWA and RWA are displayed as their respective icons. When a vehicle is destroyed during an exercise, its situation awareness information disappears.

Displaying Messages and Reports The SAFstation message log displays messages and reports from the simulated entities and units that are under its control. The contact, spot, and shell reports are generated in the message log. The log also shows orders issued to the simulated entities and units from that SAFstation.

3.7 Force Control

You can issue orders to a unit or entity under your control by clicking on the entity's icon or picture in the battlefield map.Three methods of control are available: preplanned, immediate intervention, and reactive.

Preplanned and immediate intervention control are always available. Reactive control is internally issued by the ModSAF simulation when the simulation detects a predefined battlefield situation. You can, however, specify the contingencies to which the units should react as well as the actions to take when reacting.

3.7.1 Preplanned Control

Preplanned control supports:

- Missions, or frames, and the units that perform the missions.
- Control measures such as phase lines, travel routes, and points.
- Conditions that determine whether to terminate or transition between mission phases (such as when the specified unit reaches a phase line or the end of its route).

Selecting Unit of Interest Select the unit of interest during mission assignment by clicking the mouse button on the unit displayed on the map or in a hierarchical graph of units. Once you have selected the unit, the types of missions (frames) the unit can execute are presented in a taskframe pulldown menu. For example, selecting an M1 platoon results in a menu of frames such as "Move" and "Assault."

When you select a frame assignment, the SAFstation prompts to fill any required parameters to that frame's tasks. Within each mission or mission phase, you can redefine behavior for the unit or entity by setting the mission's task parameters (such as speed and formation). Note: Some parameters are entered using the keyboard; others are entered using graphics.

It is possible to issue commands to a subordinate unit while its superior unit is executing its mission. Without further user interventions, the subordinate unit performs its mission while the superior unit adjusts its formations and actions in tactically realistic ways to accommodate the loss of the subordinate unit. The subordinate unit can rejoin its superior unit when directed.

3.7.2 Immediate Intervention Control

Immediate intervention control lets you issue commands interactively to override either preplanned or reactive behaviors. This GUI is optimized to provide maximum speed for a few functions needed frequently by users. You can either override previous commands or temporarily suspend them until the immediate intervention command is completed.

Parameters under immediate control include: Speed, Formation, Fire Permission, Target Priorities, Resume, and Task. You can also select the following (when applicable for the entity): Assault, Change ROE, Advance to Position, Withdraw to Position, Attack by Fire, Orbit, Go to Point, Flight Altitude, and Land orders.

3.7.3 Reactive Control

Reactive control lets you:

- Specify unit reactive behavior to a set of situations and battlefield events to which the units respond (e.g., enemy attack)
- Modify the parameters of those situations and events (e.g., attack from a high or low threat)
- Specify the unit's reaction to each situation or event (e.g., perform an assault).

3.8 Stealth Control

An icon on the map indicates the location and viewing direction of each Stealth (Flying Carpet) on the network. The SAFstation enables the following operations with any Stealth on the simulation network:

- Teleport the Stealth to a location on the database.
- Attach the Stealth (using attachment modes) to any entity under its command.

3.9 Simulation Events

Using SAFstation, you can interactively create artillery bursts in any location or area on the terrain. Various options are available: number of rounds, dispersal pattern and distance, rate, round types, and time delays between rounds. Bombs, mortars, howitzers, and MLRSs are also available options.

Note: It is not yet possible to create multiple missions so that artillery can fall at a specified time during an exercise; only immediate fire missions can be performed.

In addition, you can use the SAFstation to create and modify minefields. Whenever a breach lane is specified, the minefield is cleared of all mines within a specified distance of the lane.

Note: ModSAF V1.5.1 provides improved reactions and notifications when ground vehicles encounter minefields, artillery, and general indirect fire.

3.10 Message Log

The SAFstation message log displays status messages and reports from the simulated entities and units under its control. These messages and reports include contact, spot, and shell reports.

The message log displays orders issued to the simulated entities and units from that SAFstation. The messages in the message log are timestamped. The message log can be saved to a disk file.

Note: Radio messages display only for vehicles that match the local force settings or that are commanded by the workstation that controls the vehicles.

3.11 HHour Time

The SAFstation sets an HHour time enabling ModSAF entities to perform coordinating actions. The HHour editor lets you assign a name to a particular time. You can then reference this name in all phases of a mission.

This editor creates an HHour object that contains the name, date, time, defined flag, and an indication of whether this time is defined for use by friendly or enemy vehicles. These HHours can be referenced during phase transitions to start the next phase at a particular HHour, or at an offset of a particular HHour.

3.12 Resupply

The SAFstation provides a method for setting the fuel levels and weapons loads of entities at resupply locations. You specify the resupply locations and save the overlays. Logistics vehicles resupply ground entities.Protocol support for combat service support (CSS) allows supply protocol data units to communicate through the network. This capability is integrated with the Unit Supply, Entity Supply, and Entity Receive tasks. These tasks can refuel a specified group of vehicles by the US M978 Fuel Supply HEMMT.

The M978 interactively refuels vehicles within a certain radius of a chosen destination. For example, an M1 platoon running low on fuel can be refueled by a M978 truck.

Note: The Battlemaster resupplies any SAF unit or entity at any time.

3.13 Status Panel

The SAFstation has a status monitor, or panel, that displays a continuous update of the mission.

3.14 Mode Control

There are three privilege levels:

- System Operator - Highest level
- Battlemaster - Mid-level
- Commander and Operator - Lowest level

The SAFstation supports several user privilege levels. (Note that it requires a password to move the SAFstation to a higher privilege mode.)

The highest privilege level mode, System Operator, provides the most functionality since it does not disable any SAFstation functionality.

The middle privilege level mode, Battlemaster, provides more functionality than the Commander mode but less than System Operator. To edit a ModSAF unit parameter, the SAFstation must be at the Battlemaster level or higher. This means that the Battlemaster, but not Commander, can teleport a vehicle or assign the level of proficiency for a unit.

You can perform the following operations while in Battlemaster mode:

- Delete all units and missions in one PO database with one operation
- Place units on the terrain for simulation
- Retrieve scenarios from disk files
- Perform edits (such as setting the supply and skill levels) on an entity or unit
- Teleport groups of units and entities to new locations
- Set SAFstation alignment
- Delete units and entities

System Operator mode lets you perform file operations and also provides Battlemaster mode functionality.

Note: A filter controls what displays–whether you see what the forces see or only what some subsets see. This filter is optional in System Operator and Battlemaster mode but mandatory in Commander mode.

Commander mode can run a preloaded scenario and control the commander's ModSAF entities in that scenario only. In Commander mode, the map shows only the entities on the same side as the entities controlled by the commander, and enemy entities detected by the commander's entities.

3.15 Scenario Usage

A scenario file contains information about entities, units, unit hierarchies, missions, tasks, and control measures that are currently controlled, or created, by a SAFstation. This file allows a new exercise to be initialized at the point where the previous exercise was saved.

The ModSAF system allows the Battlemaster to load a scenario file. This capacity to store, edit and retrieve exercises facilitates repeated engagements.

A ModSAF commander plans a mission on the map by drawing graphics (phase lines, routes, and objectives) and selecting units to carry out missions. This mission can be created prior to an exercise and stored in a scenario file.

4 SAFsim

The SAFsim simulates and controls ModSAF entities and units. It replicates outward behavior of units and their component vehicle and weapon systems to a level of realism sufficient for training and combat development. The kinematics and dynamics of its vehicles are indistinguishable by soldiers in manned simulators from those expected of manned simulators. The tactical behavior of all semiautomated units is designed to be doctrinally correct with defaults drawn from unclassified sources.

Controlling ModSAF Behavior

ModSAF behavior is controlled by taskframes–a collection of related tasks that run simultaneously. A task is a behavior performed by a ModSAF entity or unit on the battlefield. A set of representative tasks is defined by their characteristic parameters. Default values, drawn from standard military doctrine, are provided by the ModSAF system for the task parameters. These default values can be modified. Taskframes are typically composed of move, shoot, and react tasks. Taskframes are defined in a data file that contains their component tasks and the parameter values for those tasks.

Mission

A mission is a network of taskframes connected by enabling tasks. An enabling task determines when a condition has been met so that a unit can transition between mission phases. Examples of enabling tasks are the watching for a unit to arrive at a control measure such as a phase line, or watching for user authorization.

The information included in a mission includes, but is not limited to: formations, locations, and contingency planning (for example, what to do when under attack). The tasks in a mission have intelligent default values for many of their parameters; however, the ModSAF commander can override these defaults.

Reactive tasks can interrupt the execution of a unit's current frame. These interruptions can be automatically generated by tasks within the currently executing frame or by a superior unit to modify the execution of the mission.

Tasks and missions are transparent to, and modifiable by, the commander.

4.1 Exercise Control

When you request simulation of entities or units, the SAFsim receives creation requests through the PO database. A SAFsim on the simulation network, running with the same simulation exercise and PO database as the requesting SAFstation, can perform the simulation. If there are multiple eligible SAFsims, the SAFsims determine which one should perform the simulation. The selected SAFsim creates and controls the entity until told to delete or transfer it.The SAFsim updates the simulation and PO databases whenever the state of the entities that it is simulating changes. The update information includes the position, damage level, amount of fuel and ammunition, and marksmanship level.

The SAFsim can delete an entity that it is simulating; however, only the SAFstation controlling the entity can request its deletion.

Each SAFsim monitors the state of all other SAFsims on the network. If a SAFsim drops from the network, the remaining SAFsims determine which one controls the missing SAFsim's entities. The load is balanced among the remaining SAFsims to the best possible extent.

4.2 Entity Simulation

A ModSAF entity executes a realistic range of basic commands to model the entity's inherent behavior. Some of the entity behaviors that the simulation software accurately models includes: vehicle dynamics, ballistics, battle damage, obstacle avoidance, and targeting. In addition, resource depletion and resupply is accurately simulated for fuel and ammunition.

ModSAF entities exhibit mobility, firepower and communications combat damage according to type of weapon used, location and angle of hit, incidence of hit, and range of weapon. In addition, weapons systems exhibit realistic rates of fire and realistic trajectories.

ModSAF vehicles automatically attempt obstruction avoidance, including avoiding other vehicles, and automatically perform terrain-dependent micronavigation. For example, an order from the ModSAF commander for a platoon to cross a bridge is completed without that commander having to micromanage the individual vehicles. However, collisions can and do occur. A command line argument can be used to disable drooping of gun on firepower kill and canting of vehicle on mobility kill. The visual fidelity of simulated vehicles and weapons systems is equivalent to the fidelity of manned simulators and associated weapons systems. This includes the firing flyout and impact signatures of all weapons.

4.2.1 Hull Simulation

The following hull dynamics models are available: tracked ground vehicle, wheeled ground vehicle, dismounted infantry, fixed wing aircraft, rotary wing aircraft, and missiles.

Vehicles are created containing the amount of fuel based on the vehicle's parameter database or the amount specified when creating the vehicle at the SAFstation. A vehicle's fuel consumption rate is determined by its vehicle dynamic model and fuel consumption parameters in the vehicle parameter database. The fuel consumption rate is reduced when a vehicle is idling. Vehicles are resupplied from the SAFstation by the Battlemaster.Vehicles and DI can turn and move toward a goal point. They can also travel a route consisting of sequenced multiple points. A ground route can contain either cross country or road segments depending on the order. Routes generated for ground entities avoid lakes and uncrossable river segments.

4.2.1.1 Tracked Ground Vehicles

Tracked ground vehicles can move forward, backward, and turn in place. Their orientation is determined by their direction and underlying terrain. Environmental factors, including slope and terrain type, are considered during movement.

4.2.1.2 Wheeled Ground Vehicles

Wheeled ground vehicles can move forward and backward. These vehicles, which have a minimum turn radius, must be moving to turn. Their orientation is determined by their direction and underlying terrain. Environmental factors, including slope and terrain type, are considered during movement.

4.2.1.3 Dismounted Infantry

ModSAF Dismounted Infantry (DI) can move and turn in place, and each soldier can carry and use a weapon whenever a line of sight exists to a target. Configuration information for DI is provided in the vehicle parameter database.

Each DI soldier can be in one of three postures: standing (in place or moving), kneeling, or prone. DI are vertical when standing or kneeling. Their orientation when prone is determined by the underlying terrain.

Infantry can mount appropriate vehicles (such as IFVs), ride to another location, and dismount. While mounted, DI are not visible.

Note: In ModSAF V1.5.1, teams can consist of two of more individual DI entities. One member of a team is configured with an AT missile weapon that can fire at tanks or aircraft (STINGER, SA-16). In addition, teams can move and/or "keep station" with each other.

4.2.1.4 Fixed Wing Aircraft

Fixed wing aircraft (FWA) have six degrees of freedom. The FWA hull model calculates lift, drag, and thrust. Effective limitations on roll, pitch, and yaw rates and accelerations are enforced. ModSAF FWA dynamics uses flight data from Height-Mach (H-M) diagrams to determine the power available at any point in flight. Flight data from V-N diagrams are used to enforce aerodynamic and structural limits on lift at any point in the flight envelope.

4.2.1.5 Rotary Wing Aircraft Hull Simulation

Rotary wing aircraft (RWA) have six degrees of freedom. Its effective performance limits include maximum roll, pitch, and yaw rates, maximum speeds, accelerations, and climb rates. These limits are expressed in more basic parameters such as maximum lift.

4.2.1.6 Missiles Hull Simulation

Each type of ModSAF missile has a parameter file to outline its performance characteristics. These characteristics include targeting methods, warhead types, flight dynamics, flight characteristics, sensors, modes, limitations, and movement within the dynamics model. Both ballistic and battlefield missiles are available.Tactical missiles can use any of the following guidance algorithms configured in the parameter files: lead-pursuit, pure lead, and pure pursuit. The weapon specification also indicates fusing distance, minimum effective distance for damage, and maximum tracking angle (outside of which target tracking is lost).

You can define some aspects of the launch sequence of ballistic missiles. If the missile capabilities include getting to a specified point from its launch position, a supplied target point automatically controls the boost phase behavior for the missile. Parameters that can be specified for ballistic missiles include initial and burnout mass, roll factors (such as angle and start time), thrust, and re-entry characteristics.

Missiles can be targeted and destroyed by weapon systems designed specifically for destroying missiles in flight.

Missile Types

ModSAF includes the following missile types:

- Ground-to-Ground missiles (similar to the TOW missile) have a slight superelevation angle and a measurable initial velocity upon firing. Whenever a line of sight exists between a firing vehicle and its target, the missile uses it to fly directly toward the target. A missile flies over encroaching terrain whenever there is partial line of sight from the firing vehicle to the target. A missile can coast after the powered portion of the flight is finished.

- Ground Air-to-Ground Missiles (similar to the Hellfire missile) are fired with a slight superelevation angle. The missile flies along its initial trajectory until it finds a lased target point to track to. It then flies toward the target in the X Y plane, climbing until it achieves a predetermined angle between the direction of travel of the missile and the direct vector to the target. Maintaining this angle to target slowly pulls the missile flight angle down as the missile approaches the target. Once the missile passes into a conical area above the target point, it flies directly at the target point until impact.

  This missile flight pattern occurs when there is a line of sight between the vehicle designating the target and the target, and also between the target and the missile. The missile need not be powered during the entire flight. Note: the vehicle that fired the missile does not have to be the vehicle designating the target.

- Long Range, Radar Guided, Air-to-Air Missiles (similar to the Phoenix missile) are launched by the firing vehicle, and fly straight to the target vehicle using lead pursuit guidance.

  The firing aircraft guides the missile. When the target is within the tracking capabilities of the Phoenix missile, the missile turns on its onboard radar and tracks itself to target, independent of its firing aircraft. The firing aircraft must be available to command the missile into self-tracking mode, otherwise the missile cannot lock onto a target.

- Medium Range, Radar Guided, Air-to-Air Missiles (similar to the Sparrow missile) fly toward their targets using lead pursuit guidance as long as the firing aircraft illuminates the target with its radar. A position change by the firing aircraft can cause the missile to lose tracking as when a firing aircraft turns away so that the target is no longer radar illuminated.

- Short Range, IR Guided, Air-to-Air Missiles (similar to the Sidewinder) must be locked onto a target before firing. They fly directly toward a target using pure pursuit guidance as long as there is a line of sight between the missile and the target. Once fired, the missile is self tracking (IR seeking).

4.2.2 Turret Simulation

ModSAF vehicles can have turrets that rotate, elevate, and depress any mounted weapon to a specified limit. Turrets scan to track targets, and occasionally scan to a different position within the vehicle's main arc of observation when the vehicle is not engaging an enemy. Turret parameters (scan rate to a position, scan limits, and the position offset to the attached hull) are defined in the vehicle parameter database. Battle positions are defended with turrets in down positions.

4.2.3 Weapon System Simulation

Each type of weapon system is specified in the parameter database. This specification includes: weapon position on the vehicle, amount of ammunition available to each weapon, prefire time delays and weapon fire rates, mount information (turret or hull), and weapon range (area around the vehicle that the weapon can be brought to bear on).Weapon types include missiles (visible during flyout and produce direct fire), indirect fire weapons (invisible during flight), direct fire weapons, and MLRSs.

Missile Launchers

Missile launchers in M2, BMP1, BMP2, T80, MARDER, and JAGUAR1 are replaced with ballistic guns doing flyout and using the same munitions as were in the missile launchers, for example, non-ballistic munitions. Flyout can also be shown on ballistic munitions although the munitions are traveling at a speed that is difficult to see on the Stealth.

Direct Fire Weapons

With direct fire weapons, a hit model determines if the round from the weapon strikes the target. The hit model is influenced by the weapon being used, the firer's velocity and range to target, the target's vehicle type, aspect angle, velocity and percent exposure (visibility). These factors determine if the shot was a hit or miss.

Turret-Mounted Weapons

Turret-mounted weapons require the turret to track and elevate relative to the direction of the target. The orientation of turret-mounted weapons changes since turrets occasionally scan through an arc in front of the vehicle or through an arc determined by the vehicle's position in the unit formation. Turrets can scan only when the vehicle on which they are mounted is both "alive" and firepower enabled.

Firing Weapons

Weapons can be loaded before acquiring a target. This allows vehicles to fire any weapon without waiting for the one just fired to reload. Given a target, a weapon system places the weapon in the correct firing position, fires the weapon, and reloads the weapon. Appropriate delay times are enforced. When appropriate, vehicles stop to fire if required (for example, the M2 must stop when firing a TOW).

Permission to Fire

You determine the situations in which an entity can fire its weapons. Overall fire permission can be withheld or granted. Unless specifically directed to fire at a location with no targets, firing occurs only when a particular target is considered within the range specified. Entity firing is limited by the distance limitations of the weapons systems of that entity.

Gun Behavior

Low-fidelity models of generic ballistic gun behavior and generic indirect fire ballistic gun behavior are provided for tank main guns and machine guns. Examples include 2B11, M198, M224, and M252 guns.

These guns support burst shooting, multiple types of munitions, and table-driven hit probabilities. Also, the ability to use a fire and forget tank munition (Smart Target Acquisition Fire and Forget (STAFF) round for M1A1 and M1A2 tanks) is available. The capability to hit unintended targets is also supported.

Radio messages between artillery are handled through an artillery radio model. This artillery traffic appears in the SAFstation message log window. Radio messages appear only for vehicles that match the local force settings, or that are commanded by the workstation that controls the vehicles.*Indirect Fire Simulation*

Indirect fire simulation includes the proper orientation of the gun and/or vehicle for particular fire missions. An FDC (fire direction center) model running on a vehicle processes and sorts various calls for fire and then distributes fire commands to the available artillery units. A Fire Direction Control task lets artillery register with an FDC. This FDC task can handle up to 24 vehicles. Firing patterns, munition, and number of rounds are selected by the FDC in response to the target type and size. Artillery-delivered minefield firing points are also selected.A vehicle task, running on the artillery vehicle, processes artillery radio fire requests. Fire requests to an artillery vehicle

can be sent from the SAFstation. In addition, fire orders can be sent to registered units through the SAFsim parser.

Note: Doctrinal scanning which defines the scan sectors is operational.

4.2.4 Minefield Simulation

Minefield simulation provides appropriate mine response to a given target by using search algorithms to determine targets.

You can order artillery and MLRS vehicles to fire mine dispensing rounds at an area. The detonation of these rounds cause minefields. Artillery round detonation points are simulated with errors found in the field.

Mine Marker

Mine Marker creates and simulates breached lanes through minefields with visible indicators that show a clear path.

A tool for creating Minefield Markers and Breached Lanes is available on the GUI. When you select the Mine Marker icon, the Mine Marker editor appears. Once the Mine Marker or Breached Lane is clear, clicking on an object in the PVD causes the Mine Marker editor to display letting you modify the simulation.

4.2.5 Sensor Simulation

You can specify sensors on each entity. Sensor types include: visual, infrared, and radar. Parameters for each sensor type include ranges, update rates, arcs of primary and secondary operation, blind areas, effective distances, and chances of detection.

Sensors determine what a vehicle can and cannot detect in the environment. Target entities are tracked if the following criteria exist:

- The entity is determined to be within the sensing capabilities of a tracking vehicle.

- The target entity passes tests that determine it is detectable.

Probability tables, based on orientation, situation, entity type, line of sight, angle of incidence are used to determine target detectability.

4.2.5.1 Radar Model

The radar model calculates the radar cross section of a target based on the target type, range to target, and target aspect angle. If the radar cross section of a target is greater than a threshold based on the given radar's capabilities and mode, the radar detects that target.

The radar model implements the following radar modes (similar to those in the F-14D AWG-9 radar): Pulse Single Target Track (PSTT), Pulse Doppler Single Target Track (PDSTT), Track-While-Scan (TWS) Manual, and Track-While-Scan (TWS) Auto. The PSTT mode is also used for the implementation of the long range missile radar.The radar model issues the radar protocol data unit (PDU) defined in the DIS standard whenever an aircraft or missile radar is on. An interim PDU based on the SIMNET Radar PDU is used until this PDU is standardized in the DIS Protocol Standards.

*Note:*In ModSAF V1.5.1, a ground-based sensor unit sends reports of enemy aircraft detected on long range radar. Responses to radio reports include orienting sensors and increasing the level of threat on the reported aircraft when a report is confirmed.

4.2.5.2 Visual Model

The visual model sensor detects effective target size and occlusion (actual size, aspect angle, range, percentage visible), the eyepoint of the viewer, the probability of detection, and the focus of attention. The models are easily extended to support different illumination and weather conditions. The visual subclass not only models visibility calculation but also the NVEOL detection model. A vehicle can have up to eight (8) (an arbitrary constant) visual sensors, each with its own set of unique parameters.

4.2.5.3 Infrared (IR) Model

The IR model is similar to the visual model but includes calculations for IR signature strength

rather than effective target size. Factors for IR signature strength include thrust levels for aircraft, aspect angle, and entity type.

4.2.6 Damage Simulation

Damage models (specified by the user) define how particular weapons impact a particular entity. The following factors affect this model: angle of incidence of the impact, which component of the target is hit, and the number of rounds taken. Missiles are subject to damage by specifically targeted antimissile systems.

4.2.7 Collision Simulation

Collision simulation provides a 3D physical model of collision detection. It can detect collisions with other network entities (platforms, missiles, and structures) as well as treelines, buildings, and the ground. It also generates and processes collision PDUs.

A parametrically controlled timing heuristic is used to filter out redundant collisions, such as when both parties in the collision send one another collisions PDUs.

4.2.8 Entity Projections

The following entities are available for projection onto the simulation network. Projection onto the network occurs if the specified simulation protocol is supported.

4.2.8.1 US Entities

US (American) vehicles and individuals that you can simulate include:

```
M1v
 Abrams main battle tank
M2 Bradley infantry fighting vehicle
M3 Bradley cavalry fighting vehicle
M1A2 Abrams main battle tank with a thermal (IR) commander sight
M106A1 tracked mortar carrier
M109 self-propelled 155mm howitzer plus the M109A1, M109A3, M109A5
     and M109A6
```

```
M113 ambulance; armored personnel carrier - ambulance vehicle
M113 observer; armored personnel carrier - observer vehicle
HMMWV high mobility multi-purpose wheeled vehicle
M88A1 medium tank recovery vehicle
M977 HEMTT-Cargo; heavy expanded mobility tactical truck (8 ton cargo)
M978 HEMTT-Fuel; heavy expanded mobility tactical truck (2500 gal)
DI dismounted infantry
A10 Thunderbolt II; ground attack fixed wing aircraft
F14D Tomcat FWA; Navy long range interceptor
F16D fixed wing aircraft
OH58D Kiowa Warrior; armed scout/observation helicopter
AH64 Apache attack helicopter (with Hydra submunition rockets)
Commanche (next generation armed scout/observation helicopter)
Avenger air defense
M2 Stinger air defense
M270 GAT2; a MLRS preloaded with German AT2 rockets (mine subs)
M270 M26 DPICM; a MLRS preloaded with M26 rockets
M270 M77 extended range DPICM; a MLRS preloaded with M77 rockets
M270
M981; a US M113 variant configured as a Fire Support Team Vehicle
M992
```

American missiles that can be simulated include the Phoenix, Sidewinder, Sparrow, Maverick, and TOW missile.

4.2.8.2 Russian Entities

You can simulate the following Russian entities:

```
T80 medium battle tank
T72 medium battle tank
BMP1 armored fighting vehicle
BMP2 armored fighting vehicle
Mi24 HIND attack helicopter
MIG27 Flogger fighter aircraft
MIG29 Fulcrum fighter aircraft
SA9 short range air defense missile launcher
SU25 Frogfoot ground attack aircraft
ZSU23-4 self-propelled quad 23mm air defense vehicle
2S1
2S6
BRDM2
BTR80
BTR 60PU artillery vehicle
URAL 375C combat support vehicle
URAL 375F 5.0 ton cargo / fuel truck
```

```
DI dismounted infantry
1V13 Btry FDC
1V14 Btry COP
1V15 Bn COP
1V16 Btry FDC
2B11 wheel mounted, towed mortar
2S19
BM21
ZIL131 FDC
Mi28 RWA
Mi8 RWA
```

Russian missiles that can be simulated include the Archer, Alamo, Songster, SA-19, Spigot, and Spiral missile.

4.2.8.3 German Entities

You can simulate the following German vehicles and personnel:

```
LEO1A5 Leopard IA5 medium battle tank
LEO2 Leopard II medium battle tank
MARDER1A3 armored fighting vehicle
MTW M113 observer vehicle
JAGUAR1 and JAGUAR1-2
SKORPIAN
PAH1 helicopter
DI dismounted infantry
```

The German missiles that can be simulated include the Milan and HOT missile.

4.2.9 Entity Tasks

Behaviors are categorized into hierarchical tasks. Tasks use entity, environmental, and internal state to generate control inputs that guide the entity in accomplishing its mission. Tasks applicable to most entities follow:

4.2.9.1 Assess

The Assess task identifies the most urgent target and recommends which weapons to use against

that target. The task takes its primary input from the Spotter task which provides lists of vehicles detected by available sensors.There are two firing types: distributed and volley. For distributed firing, the Assess task tries to choose a target that is not targeted by someone else. If all detected enemy vehicles are already targeted, the vehicle targets the highest priority enemy vehicle.

For volley firing, the Assess task chooses a target that is the highest priority. This firing type does not check if the detected enemy vehicle is already being targeted.

Selecting Target The Assess task selects the next target to attack. Targeting characteristics match weapons and targets to select the best target for the situation. Selection of best target is based on weapon capabilities, weapon priorities and target type, distance to target, ammunition availability, target priority, and weapon enabled lists.

Multiple Targets When multiple targets are available for a vehicle, target selection chooses the most appropriate one. You can create a target priority list based on a set of vehicle types. Available target classes include tanks, command vehicles, APCs, rotary wing aircraft, artillery, anti-aircraft vehicles, logistics vehicles, fixed wing aircraft, ships, missiles, and dismounted infantry.

Weapons data for the Assess task, read from an entity's parameter file, specify which weapons and munitions are recommended against a threat. Groupings of data by weapon type is provided for different types of targets. For example, weapons data may be used against a tank, a ground vehicle that is not a tank, and a helicopter.

In addition, the weapons data for a tank threat may specify using a main-gun Sabot round as the first priority weapon against a tank threat (whose range is between 0 and 3500 meters), or a main-gun Heat round if no Sabot rounds are available.

Munition Choice

The weapons data lists munition choices. The recommended munition is the first for which the threat falls within the *min* and *max* ranges, and the detection level for the target matches or exceeds the specified detection level.

The ModSAF detection levels are: detect, classify, recognize and identify.

4.2.9.2 Enemy

The Enemy task accumulates incoming rounds near a ground entity to determine if the ground entity is under attack. The task maintains information about the number of rounds and the firer.

Query Interface Functions

Query interface functions enable other functions to determine if the vehicle is under attack, and if so, by whom. This task also maintains information about the number of times an entity has successfully hit a target with a munition.

4.2.9.3 Enemy Detection

The Enemy Detection task identifies the enemy by maintaining a list of the enemy vehicles that can be detected by vehicles in a unit. Minefields, incoming artillery, enemy superiority, high casualties, and air raids are also detected.

4.2.9.4 Spotter

The Spotter task accumulates detected vehicles from a vehicle's sensors. An IFF (Identification, Friend or Foe) model for visual sensors is included.

This task provides a list of detected entities to other tasks within this entity or to other entities and applications (such as the user interface).

Parametric Data

The parametric data for the Spotter task includes a list of the entity's sensors, the frequency with which the task should provide a network update of detected vehicles and their presumed alignments, and the time a target remains on the detected list after it is no longer detected by a sensor. This last parameter helps to implement a simple model of crew memory.

4.2.9.5 Search

The Search task tries to point a sensor either at a given location or at a set azimuth and elevation

relative to the host vehicle. It can be used in a taskframe to support searching for enemies with a sensor. This task also supports turret slewing to simulate the turret scanning actions in ground vehicles with turrets.

An entity's parametric data can specify the type of search (such as ground or air), and turret slewing parameters (such as slew rate and time to pause at search sector boundaries).

4.2.9.6 Targeter

The Targeter task receives threat information from a recommendation function and performs targeting actions against that threat. The targeting actions are hull, sensor, turret, and gun commands. *Note:* Weapons cannot shoot if the fire permission is disabled.

Parameters, specifying each munition/weapon pairing that is available during an engagement, classify munitions as follows:

`'Ballistic'`
: A round requiring no tracking by the entity during the round's flight. An example is the main gun of the M1 tank.

`'Fire-and-Forget'`
: A missile requiring no tracking by the entity during the missile's flight. An example is the Sidewinder missile on the F14D airplane.

`'Host-Tracking'`
: A missile that requires tracking by the host entity's radar system for the missile to lock on to the target. An example is the Sparrow missile on the F14D airplane.

`'Host-Tracking-with-Onboard-Radar'`
: A missile that requires tracking by the host entity's radar system for the missile to lock on to the target during the first phase of flight. The missile has an onboard radar that can be used during the later portions of its flight, thus freeing the host entity from radar tracking during the final phase. An example is the Phoenix missile on the F14D airplane.

`'Host-Steering'`
: A missile that requires steering commands issued by the host entity for the missile to track the desired target. An example is the TOW missile on the M2 IFV.

`'Host-Steering-with-Onboard-Radar'`
: A missile that requires steering commands issued by the host entity for the missile to track the desired target during the first phase of flight. The missile has onboard radar

that can be used during the later portions of the flight, thus freeing the host entity from issuing steering commands during the final phase.

When a munition/weapon pair requires tracking by the host, task parameters specify the host's tracking component (gun or sensor). If the sensor component supports modes, the following parameters are included:

- The desired half-width and half-height of the radar tracking volume. These are specified in degrees, and imply a volume containing plus and minus the specified number of radians about a center beam pointing at the target.

  Note: As of V1.5.1, the ground-based sensor unit sends radio reports of enemy aircraft detected on long range radar.
- Minimum and maximum desired velocity thresholds for the sensor.
- The desired range, in meters, of the sensor tracking volume. This number must be large enough to reach the target during the tracking phase of an engagement.

When a missile has an onboard sensor, task parameters identify the sensor and the sensor mode that should be used to lock on to a target. The range, from a target at which the missile's onboard sensor should be turned on, is provided when the missile requires host tracking or steering.

4.2.10 Ground Entity Tasks

Ground entity tasks are described in the following sections.

4.2.10.1 Move

The Move task generates commands to the hull component directing it to move an entity along a route and around obstacles.

An entity can selectively avoid rivers, lakes, buildings, trees, tree lines, and other vehicles. This task determines when the entity should get on a route. Ground vehicles automatically avoid obstacles they encounter in their paths. The direction of travel and speed of other vehicles is calculated to ensure avoidance. This task is a ground vehicle-specific task.

4.2.10.2 Collide

The Collide ground entity task allows vehicles to recover from collisions and near-collisions. It directs colliding entities to wait a random amount of time and then back out. This task is also invoked when a vehicle cannot move around an obstacle.

4.2.10.3 Terrain

The Terrain ground entity task contains a local terrain map that vehicles use during route planning. This map also displays local obstacles that entities should avoid.

4.2.10.4 Stingray React

The Stingray React ground entity reactive task simulates permanent damage received from a stingray-equipped simulated vehicle on the network. If a vehicle receives permanent damage due to Stingray fire, it searches for a hidden position in the surrounding terrain.

An entity's parametric data for this task determines whether to check for a Stingray hit and, if the vehicle is damaged, the speed at which it should move to a hidden position (in meters/second). Currently the following vehicles check for a Stingray hit: M1, M2, T72, BMP1, and BMP2.

4.2.10.5 Mount

The Mount ground entity task mounts DI onto its corresponding Infantry Fighting Vehicle (IFV). When the DI is dismounted, the Mount task waits for the appropriate IFV to get within a certain range. After a set time, the DI disappear. The DI are then considered mounted and the Mount task is in the mounted state.

When the Mount task is in the mounted state it waits for a dismount request. After a set time, the Mount task causes the DI to reappear in a location behind the IFV. The task returns to the dismounted state.

4.2.10.6 Indirect Fire Mission

The Indirect Fire Mission task processes artillery radio fire requests for artillery vehicles. The gun fires the artillery. Vehicles that are running this task with turret traverse limits must adjust their hull orientation to the target line if the turret azimuth is out of range.

4.2.10.7 Supply

The Supply task implements resupply of a vehicle. The task processes the necessary issue and reception of supply protocol packets.

4.2.10.8 Receive

The Receive task requests supplies from a resupply vehicle if:

- A resupply vehicle is next to the vehicle
- The resupply vehicle is halted
- The vehicle is halted
- The vehicle needs supplies

4.2.10.9 Backtrack

The Backtrack task implements a vehicle-level task that causes a vehicle to backtrack for a given distance. This is done by retrieving the history list (a list of points where the vehicle has previously been), reversing it, and setting the vehicle to move backward along the route described in the reversed list.

4.2.10.10 MLRS

The Multiple Launch Rocket (MLRS) task implements the firing and mission acknowledgement characteristics of Multiple Launch Rocket vehicles. It loads the ammunition included on the vehicle, waits until the vehicle is in firing position, and then fires. At present, vehicles running this task are considered "one shot"; their munitions are expended during the mission. *Note:* A resupply structure is not yet in place.

4.2.11 RWA Entity Tasks

Behaviors are compressed into hierarchical tasks. Tasks use entity, environmental, and internal state to generate control inputs that guide the entity in accomplishing its mission. RWA entity tasks are described in the following sections:

4.2.11.1 Fly Route

The Fly Route vehicle-level task follows a route. It is capable of following the waypoints of a line, or of going directly to a point or text. Treelines, canopies, and buildings are considered when contouring RWA flight; you can safely create vehicles in tree canopies or given routes with waypoints at treelines.

4.2.11.2 Land

The RWA Land vehicle-level task flies the RWA to a landing area and lands. The RWA stops, faces a given direction, and descends.

4.2.11.3 Hover

The Hover vehicle-level task implements hovering of rotary wing aircraft. The hover altitude and direction can be specified. The RWA stops, achieves the specified altitude above-ground level and faces toward the specified direction.

4.2.11.4 Orbit

The RWA Orbit vehicle-level task implements an orbit flight pattern around a point.

4.2.11.5 PopUp Attack

The RWA PopUp Attack vehicle-level task aligns an RWA aircraft with a desired direction and forces it to ascend to a minimum PopUp altitude. When the minimum altitude is reached, the

PopUp Attack task causes the RWA to search for enemy vehicles while it continues to climb to the maximum PopUp altitude.

If an enemy vehicle is detected during the ascent, the ascending maneuver terminates and the aircraft holds its current altitude. This behavior helps the RWA engage against the enemy vehicle. After a set period of time, the PopUp Attack task forces the RWA to return to the initial altitude. However, if the maximum PopUp altitude is reached before detecting enemy vehicles, the PopUp Attack task orders the RWA to return immediately to its initial altitude.

4.2.11.6 Running Fire Attack

The RWA Running Fire Attack vehicle-level task implements the running fire attack maneuver. It spawns the RWA Run task on the vehicle and waits for the task to end.

If live targets (both ground and air) are within a user-specified radius of the attacked objective and the RWA is not out of ammunition, a Fly Route task is spawned to return the RWA to its starting point.

4.2.11.7 Run

The RWA Run vehicle-level task implements one sweep of the running attack fire maneuver. The RWA flies at a user-specified altitude searching for targets. When a target is detected, the RWA flies toward the attack objective while firing missiles at the target. The task ends when the RWA gets too close to the attack objective, the RWA runs out of ammunition, or the target is eliminated.

4.2.11.8 React to Contact

RWA React to Contact implements a unit-level reactive trigger. This task monitors enemy activity using UEnemy. Once enemy vehicles are detected, the URWAReactContact task triggers a reaction. If the unit is occupying a position and the enemy comes from the direction that the unit is facing, a URWAPopUp task is executed. Otherwise, the unit reacts in the following way:

- If the RWA unit is on the ground when the enemy is detected, it executes a Ground Drill task.
- If the RWA unit is in the air when the enemy is detected, it executes an Air Drill task.

4.2.11.9 Ground Drill

RWA Ground Drill implements the following drill that an RWA unit executes when attacked on the ground:

- If the scramble parameter is set when the task is invoked, the RWA ascends and flies quickly away from the enemy. It flies for about 45 seconds (parametric data) and then moves toward a rendezvous point.
- If the scramble parameter is not set, the RWA immediately moves to the rendezvous point.

  Once the unit reaches the rendezvous point, it attacks the enemy (if the fire permissions are not set to Hold Permission). If the fire permissions are set to Hold Permission, the unit holds at the rendezvous point. To attack the enemy, the URWAGroundDrill task spawns a URWAAttack task specified by the Reactive Trigger task.

4.2.11.10 Air Drill

Using the Reactive Trigger task, the Air Drill task implements a drill that an RWA unit executes if attacked in the air. If the scramble parameter is set when the task is invoked, the RWA breaks formation and turns either 45 degrees left or 45 degrees right depending on its position in the formation. The RWA accellerates to top speed and dives. It then executes a 90 degree turn in the opposite direction and makes another 45 degree jink.

If the parameter is not set, the RWA performs the same maneuver as above but stays in formation.

After performing maneuvers, the RWA may or may not attack the enemy, depending on the permission parameter.

RWAs do not fly from target during air drill when the leader is eliminated.

4.2.11.11 RWA Bounding Overwatch

RWA Bounding Overwatch implements a unit-level task that performs two methods of Bounding Overwatch: successive or alternate bounds.

An additional bound-type parameter specifies whether the bounding is "successive" or "alternate." With successive bounding, the functional groups "leapfrog" along the route. With alternate bounding, the same functional group travels into new territory first, and the following group "catches up."

The unit divides into two functional groups: one group travels the route; the other group executes an RWA Occupy Position task and watches for enemy vehicles.

4.2.12 FWA Entity Tasks

Behaviors are defined as hierarchical tasks. Tasks use entity, environmental, and internal state to generate control inputs that guide the entity in accomplishing its mission. FWA entity tasks are described as follows:

- Unit-level command enables Close Air Support (CAS) to select route and refuel point.
- Fire support editor lets you call into CAS.

FWA units recognize and engage air targets of opportunity as well as the occurrence of bingo fuel by returning vehicles that have reached bingo fuel levels. Also, these units reorganize the remaining vehicles in the unit, continue commanded taskframe execution, and properly avoid collisions with other air vehicles.

4.2.12.1 Follow Route

The Follow Route air vehicle task commands aircraft to follow a route. The task parameters include: a route, speed, and altitude. The route can be a point, line, or text graphic allowing the air vehicle to follow the waypoints of a line, or of going directly to a point or text. FWA can fly in contour flight mode without colliding with the ground.

4.2.12.2 Orbit

The Orbit air vehicle-level task commands aircraft to circle a point. Air vehicles perform an Orbit Hold by circling a point at a fixed distance and at a standard speed.

4.2.12.3 CAP

The CAP air vehicle-level task performs Combat Air Patrol. It enables aircraft to fly in a racetrack pattern while searching for enemy aircraft.

4.2.12.4 Land

The Land air vehicle-level task enables aircraft to perform instantaneous landing. The vehicle is teleported to the ground at its current X, Y position.

4.2.12.5 FWA Ground Attack

The FWA Ground Attack task controls the movement of an individual FWA during a Close Air Support (CAS) ground attack. It performs the vehicle attack phases of geometry, entry, and delivery.

4.2.12.6 FWA Formation Keep

The FWA Formation Keep vehicle-level task enables aircraft to stay in formation. It maintains a specified offset from the formation leader vehicle.

4.2.12.7 FWA Ground Avoidance

The Ground Avoidance vehicle-level task ensures that a vehicle does not collide with the ground. The task uses a simple algorithm; when it detects that the vehicle may crash, it causes the aircraft to move into a steep climb until it is no longer in danger.

Note: While the task attempts to move the vehicle out of danger, it cannot actually prevent the vehicle from colliding with the ground as it is limited by the dynamics of the vehicle. However, ModSAF vehicles do have the ability to avoid cliffs and steeply sloped terrains. Rather than attempting to climb such terrain, vehicles now drive around the area.

4.2.12.8 ATA Intercept

The ATA (air-to-air) Intercept vehicle-level task controls the movement of a vehicle during an air-to-air intercept. It guides the aircraft on a pure pursuit course of the enemy by allowing the aircraft to turn and fly a course that intercepts a designated target.

4.2.12.9 Commit

The Commit vehicle-level task determines when an aircraft should perform an air-to-air intercept of another aircraft.

4.2.12.10 Reset

The Reset vehicle-level task determines when to stop an air-to-air intercept.

4.2.12.11 Take Off

The Take Off vehicle-level task enables an aircraft to perform an instantaneous take off by teleporting to the take off altitude.

4.2.12.12 Return to Base

The Return to Base vehicle-level task flies an aircraft to a point designated as the base. When reaching this point, the vehicle lands.

4.2.12.13 Bingo Fuel

The Bingo Fuel vehicle-level task determines when aircraft must return to base to refuel.

4.3 Unit Simulation

A unit is any combination of entities or entities plus units. Unit-level simulation includes the creation and control of units. Most units are created in one of the following formations: wedge, line, echelon-left, echelon-right, trail, staggered-left, or staggered-right.Units are controlled by doctrinal tactics involving tasks and missions necessary to perform functions such as move, shoot and communicate, formation keeping, target detection, identification, selection, fire planning, and distribution.

These capabilities are based on, but are not limited to, factors such as range, motion, activity, visibility, arc of attention, background, direction, orders, mission, evaluation of threat, and environment.

4.3.1 Unit Types

ModSAF supports both normal and mixed units. Normal units are composed of one type of entity; mixed units are composed of more than one type. American normal units include:

Platoons M1, M2, M1A2, M109, or 106A1; each consisting of four vehicles.
Scout platoon M2; consisting of six M2 vehicles.
Companies M1, M2, M1A2; each consisting of three platoons plus two command vehicles (65 and 66).
US DI group platoon consisting of four DI groups; each group consisting of six DIs.
US DI section consisting of six US DI rifle soldiers.
US DI platoon consisting of four US DI sections.
US RWA pair consisting of two US rotary wing aircraft.
US RWA platoon consisting of four US rotary wing aircraft.
US RWA company consisting of eight US rotary wing aircraft.

American mixed units include:

M2 reinforced company consisting of three mech platoons of four M2s each, one tank platoon of four vehicles, and two M2 command vehicles (66 and 65).
M2 DI platoon consisting of one M2 platoon and one US DI platoon.
M2 DI group platoon consisting of one US DI group platoon and one M2 platoon.

Russian normal units include:

Platoons T80 or ZSU23_4M; each consisting of four vehicles.

Platoons T72, BMP1, BMP2, or BTR80; each consisting of three vehicles.

Companies T80, T72, BMP1, BMP2, or BTR80; each consisting of three platoons plus a command vehicle (66).

BRDM2 section consisting of two BRDM2 vehicles.

USSR DI section consisting of six USSR DI rifle soldiers.

USSR DI platoon consisting of three USSR DI sections.

USSR DI group platoons consisting of three DI groups; each consisting of six DIs.

Russian mixed units include:

BMP2 reinforced company consisting of three BMP2 platoons of three BMP2s each, plus one tank platoon of two T72 vehicles, plus three command vehicles consisting of one BMP2 (66) and two ZSUs (00 and 01).

BMP1 DI platoon consisting of one BMP1 platoon and one USSR DI platoon.

BMP2 DI platoon consisting of one BMP2 platoon and one USSR DI platoon.

BTR80 DI platoon consisting of one BTR80 platoon and one USSR DI platoon.

1BRDM2 1BTR80 section consisting of one BRDM2 vehicle and one BTR80 vehicle.

2BMP1 1BRDM2 section consisting of two BMP1 vehicles and one BRDM2 vehicle.

2BMP2 1BRDM2 platoon consisting of two BMP2 vehicles and one BRDM2 vehicle.

1T80 3BTR80 platoon consisting of one T80 vehicle and three BTR80 vehicles.

2T80 1BRDM2 platoon consisting of two T80 vehicles and one BRDM2 vehicle.

BMP1 DI group platoon consisting of one BMP1 platoon and one USSR DI group.

BMP2 DI group platoon consisting of one BMP2 platoon and one USSR DI group.

BTR80 DI group platoon consisting of one BTR80 platoon and one USSR DI group.

German normal units include:

Platoons Leo1A5, Leo2, Jaguar1, or Marder1A3; each consisting of three vehicles.

Companies Leo1A5, Leo2, Jaguar1, or Marder1A3; each consisting of three platoons and a commander vehicle.

LUCHS section consisting of two LUCHS vehicles.

DI section consisting of six German DI rifle soldiers.

DI platoon consisting of three German DI sections.

German mixed units include:

3Leo1A5 1Marder1A3 platoon consisting of three Leo1A5 vehicles and one Marder vehicle.

3Leo2 1Marder1A3 platoon consisting of three Leo2 vehicles and one Marder vehicle.

1Leo2 3Marder1A3 platoon consisting of one Leo2 vehicle and three Marder vehicles.

German DI group platoon consisting of three DI groups; each group consisting of six DIs.

MarderA3 DI group platoon consisting of one German DI group Platoon and one MarderA3 platoon.

MarderA3 DI group - Leo platoon consisting of one MarderA3 platoon, one German DI-group platoon and one Leo1A5 vehicle.

MarderA3 DI platoon consisting of one Marder1A3 platoon and one German DI platoon.

MarderA3 DI - Leo platoon consisting of one Marder1A3 platoon, one German DI platoon, and one Leo1A5 vehicle.

Air support units are provided for both fixed wing aircraft and rotary wing aircraft. Both attack and scout RWA are included.

4.3.2 Task Organization

When tactically appropriate, the ModSAF system creates its own task-organized units without user intervention. For example, when a platoon is performing a Bounding Overwatch, ModSAF splits the platoon into two functional subunits. These subunits are regarded as two separate organizations with separate tasks. (The first subunit may be tasked to seek cover and provide covering fire for the second subunit while the second subunit may be tasked to move to a good overwatch position.) The original platoon is reassembled once both subunits reach the end of the overwatch route.

4.3.3 Communication

The SAFsim models message communications between the ModSAF units and their commanders. Information is aggregated and messages are sent to the controlling SAFstation. Units collect and correlate vehicle sighting information, thereby creating spot and contact reports.

4.3.4 Platoon Tasks

By grouping entities into units, you can control multiple entities by giving commands to the unit alone. The unit then performs the specified commands/tasks as required. These tasks can

involve independent actions by subunits or entities. *Note:* You can also issue commands or tasks directly to a subunit or entity in a unit.

Tasks for platoon ground units are described in the following sections:

4.3.4.1 Actions on Contact

Actions on Contact is a unit-level reactive task that monitors enemy activity and reacts to contact. This task continually monitors whether the unit is under fire or enemy vehicles are detected. If either of these situations exist, the task reacts by executing an appropriate response. The reaction executed depends on the parameter values that you set. The reactions that are currently supported are:

`'Contact Drill'`
: A Unit Targeter task is added to the unit's activity. This permits the unit to continue executing its current frame while firing at the enemy (if the unit has fire permission).

`'Action Drill'`
: A unit-level Assault taskframe executes and enables fire permission. The Action On Contact task creates an assault objective at the computed enemy location. This objective is passed to the Assault task.

  *Note:*Actions on Contact is enhanced to provide realistic criteria for determining reaction to enemy contact.

`'Attack by Fire'`
: An Occupy Position taskframe containing unit Prep Occupy Position and a Unit Targeter task is implemented for the Attack by Fire task. The unit Action On Contact task creates an objective facing the enemy and chooses logical target reference points with the engagement area TRP at the enemy location. These parameters are passed to the unit-level Prep Occupy Position task.

The unit-level Actions On Contact task does not end until the taskframe that it resides in is destroyed. However, it reverts to the monitoring state if there are no enemies in sight since action is no longer required.

The Actions On Contact task is the sponsoring task for the reactive taskframes listed above. That is, the Actions On Contact task remains active even when the original frame is suspended. The reaction can then be easily monitored and stopped when necessary. Also, if the situation has changed enough to cause a different type of reaction, the Actions On Contact task stops the current reaction and starts the appropriate one.

Note: In V1.5.1, a more realistic set of criteria determines reaction to enemy contact. The reaction is determined by the vehicle class or the enemy as well as the number of enemies.

4.3.4.2 Assault

The Assault task arranges the movement and firing commands to perform an on-line attack on an objective. There are two ways this task can occur: planned and reaction to enemy activity. If planned, you enter the assault objective. If it is a reaction to enemy activity, the task occurs in reaction.In this case, the enemy location is passed in as the assault objective.Once the assault objective is known, the route to the objective must be found. If you entered the route through the user interface, that route is chosen. If not, a route that links the current position to the assault objective is generated by the system. The task then directs the subordinates to follow the route to the objective through the Traveling task.

When the unit reaches the objective, or when the starting unit strength drops below a certain user-determined percentage, the unit conducts a unit-level Prep Occupy Position task.

The Assault task computes a battle position based on the direction of the original position to the assault objective. This computed battle position should place the unit with its back to the original position and its front to the enemy location. This battle position is passed to the Prep Occupy Position task which finds defensible (covered or concealed) positions. Once these positions are secure, the Assault task executes an Occupy Position task until directed to do otherwise.

4.3.4.3 Bounding Overwatch

Ground vehicles and dismounted infantry units can perform bounding overwatch movement in which part of the unit covers the movement of another part by using terrain features as cover.

Functions include: proper handling of tasking away and rejoining, automatic fallback to "closed" formation when entering tree canopies, stopping between hops, preventing OW destinations inside canopies, smoother transitions between hops, and stopping at tactical signs.

*Note:*The RWA Bounding Overwatch function implements a unit-level task that performs two methods of bounding overwatch: Successive and Alternate Bounds. The unit divides into two functional groups enabling one group to move along a route while the other group is executing and RWA Occupy Position task and performing reconnaissance.

4.3.4.4 Resupply Capability

This task provides a routine that can resupply a unit. This routine can be tested from the parser interface.

4.3.4.5 DI Move to IFV (Mount Task)

The Mount task lets you enable the DI to move to the IFV rather than the IFV move to the DI. You can select a mode in which your decision is based on a tactical basis (depending on distance and presence/lack of enemy).

4.3.4.6 Utraveling Leader

The Utraveling task monitors the progress of a leader's subordinates when a platoon is moving. If a member of the platoon falls behind (as determined by parametric data), the leader of the platoon slows down so as not to get too far ahead of other members of the platoon.

4.3.4.7 Enemy

The unit-level Enemy task maintains a list of enemy vehicles that the unit (through all its subordinates) can detect. This list is a compilation derived from the list generated by the entity-level Enemy task running on each subordinate.

The Enemy task also detects the following events: minefields, incoming artillery, enemy superiority, high casualties, and air raids. This task calls the appropriate event handling functions (which have previously been registered as callbacks) when these events are detected.

4.3.4.8 Halt

The unit-level Halt task directs a group of subordinates to stop. When the unit is halted for an assembly, the unit forms a coil formation. When the unit is moving cross country, it stops in formation. When the unit is on a road, it stops and sends the subordinates to the left and right side of the road in a herringbone formation. The distance off the road and the stopping look ahead time is parametric data. This data is used to calculate the halt point for each subordinate. The task then tells the subordinates to move toward the desired points.

4.3.4.9 Hasty Occupy Position

This section describes:

- Prep Occupy Position
- Defend Battle Position
- ADA Prep Occupy Position
- Occupy Position

4.3.4.10 Prep Occupy Position

The Prep Occupy Position task is a unit-level preparatory task for Occupy Position that computes hiding positions and sectors of fire for the unit's entities. It finds covered and/or concealed positions along the battle position and instructs the subordinates to go toward these positions. This task ends when the subordinates have reached the desired positions.

Based on the battle position and the number of subordinates, both the number of vehicles per segment (the battle position consisting of one or more line segments) and the battle areas (areas where each vehicle searches for cover) are calculated. The subordinates are assigned positions from one end of the battle position to the other such that no vehicle crossover occurs while they are traveling to their positions. These positions are passed to the Move task for each vehicle. The unit task waits until the vehicles are in the desired positions.

4.3.4.11 Defend Battle Position

When performing an occupy position, vehicles move after detecting accurate hostile fire rather than firing from one position.

The vehicle-level task moves a vehicle between the primary firing position, the alternate firing position, and the hidden position (passed in as task parameters).

Primary and alternate firing positions are hull-defilade positions; the hidden position is a turret-defilade position. ModSAF computes and creates these positions as point objects, instructing the vehicles to go to the primary firing position. If the vehicle detects that it is receiving accurate fire, it returns to its hidden position. If there is an alternate firing position, the vehicle immediately moves toward it. If not, the vehicle waits a random amount of time at the hidden position and

returns to the primary firing position. If the vehicle receives accurate fire while at the alternate firing position, it moves to the primary firing position in the same manner as it moved to the alternate firing position. The goal is to enable the vehicle to take evasive action when it is receiving accurate fire.

If the vehicles are in an open plain and no primary firing positions are found, using alternate firing positions is not beneficial. In this case, the vehicles occupy their default positions on the battle position and do not change fire positions.

4.3.4.12 ADA Prep Occupy Position

The ADA Prep Occupy Position task is an ADA-specific unit-level Occupy Position preparatory task that computes hiding positions and sectors of fire for the unit's entities. It finds elevated covered positions along the battle position and instructs the subordinates to go toward these positions. This task ends when the subordinates have reached the desired positions.

Based on the battle position and the number of subordinates, both the number of vehicles per segment and the battle areas are calculated. The subordinates are ordered by job numbers, and are assigned positions from across the battle position to ensure that no vehicle crossover occurs while they are traveling to their positions. The unit task then idles until the vehicles are in their desired positions.

4.3.4.13 Occupy Position

The unit-level Occupy Position task maintains subordinates in an occupy position. If a preparatory task such as Prep Occupy Position or ADA Prep Occupy Position has placed the subordinates in covered positions, the Occupy Position task ensures that the subordinates remain in these positions until other orders are assigned.

Note: Vehicles are aware of enemies during execution of an Occupy Position task. The engagement points-of-focus is per vehicle, not per platoon (each vehicle is assigned an engagement point that is the center of mass of the most threatening enemy cluster). Engagement points move according to when the enemy moves. Searches are repeated as necessary on a per vehicle basis.

You can limit the frequency of searches, and specify an area or a line for the engagement area parameter.

4.3.4.14 Targeter

The Targeter task assigns sectors of fire and fire permissions to unit entities permitting unit members to coordinate their fire with multiple targets. For example, a US tank platoon of M1s presented with multiple targets can each fire at a target not already hosen by another member of the platoon. This occurs only with units that perform coordinated fire (i.e., US units) at the basic platoon level.

4.3.4.15 Pick Up

The Pick Up task sends IFVs to DI who are waiting to be mounted. The DI and IFVs are associated by their task-organized indices. Each IFV is sent to the corresponding DI with the same index. This task is normally spawned on an IFV platoon which is part of a Mixed IFV/DI platoon. However, this task can also be spawned on a single IFV or DI unit. If the unit is a single vehicle, the task spawns a Move task on the IFV and a Mount task on the corresponding DI. This task is normally called from the Mixed Mount task and ends when the IFVs reach the DI.

4.3.4.16 Overwatch Move

The Overwatch Move task enables platoons to travel routes in Overwatch mode. This task divides the unit into two functional groups: one group travels along the route; the other group executes a Prep Occupy Position task and watches for enemy vehicles. This method of travel is used during reconnaissance missions.

The "move type" task parameter specifies whether the overwatch is categorized as bounding or non-bounding. In Bounding Overwatch, the functional groups "leapfrog" along the route. In Non-Bounding Overwatch, one group travels into new territory first; the following group "catches up." The task automatically ends when the unit reaches the objective or when the unit is no longer capable of performing the overwatch movement (for example, when the number of vehicles capable of movement falls below two).

4.3.4.17 Traveling

The Traveling task moves a unit in its specified formation or directs a unit to perform a road-march.

Unit vehicles or DI move in formation relative to each other. A variety of formations are available for each type and size of unit. The commander can set the formation scale factors to adjust the size of the formations.

Units automatically adapt their formation to their situation. For example, as a unit arrives at a river bridge, the member vehicles of that unit fall out of formation as each approaches the bridge. They form a column to cross the bridge, and then return to their original formation on the other side. This behavior also occurs when units skirt terrain features, such as rivers or treelines, or negotiate passages too small to accommodate the original formation.

ModSAF ground units have the ability to follow another vehicle.

4.3.4.18 Pursue

The Pursue task is an extension to the unit Traveling task. It allows a platoon to "chase" another platoon. The task updates the pursue route occasionally to reflect the pursued unit's movement.

4.3.4.19 Dismount

The Dismount unit-level task dismounts subordinate DI from the IFVs in their unit. The DI and IFVs are associated by their task-organized indices. The index assigns an IFV to a DI.

This task is normally spawned on an IFV platoon which is part of a mixed IFV/DI platoon. However, it can also be spawned on a single IFV or DI unit. If the unit is a single vehicle then after halting the IFV, the Unit Dismount task spawns a ground entity-level Mount task on the corresponding DI changing its mount state to dismounted.

When the IFV is selected to dismount its DI, you should determine if that IFV is carrying DI. If there are DI to dismount, issue a Halt task to that IFV. The background Vehicle Mount task then initiates the calls necessary to dismount each DI.

4.3.4.20 Unit Mount

The Unit Mount task mounts subordinate DI onto the IFVs in their unit. The DI and IFVs are associated by their task-organized indices that assign IFVs to DIs.

When you issue a Mount task to the subordinate DIs, the initial action is to issue a Halt task to those DI who then wait for their associated IFVs to arrive. The background Vehicle Mount task initiates the necessary calls to mount each DI.

4.3.4.21 Withdraw

The Withdraw task moves vehicles away from the enemy, and then performs an Occupy Position until another order is given to the unit. All vehicles move quickly to the withdraw point or along the withdraw route. Vehicles are not required to keep formation.If the enemy is detected, armored vehicles move to the withdraw point in reverse gear. (Otherwise, they drive in forward gear.) When the detected enemy is no longer visible, an armored vehicle completes its movement to the withdraw location in forward gear. *Note:* Once it transitions from reverse to forward, it remains in forward even if the enemy reappears.Unarmored vehicles travel to the withdraw point in forward gear. All vehicles can fire while moving to the withdraw point (providing the vehicle has munitions); however, vehicles do not shoot "stop to shoot" weapons (such as the TOW missile) while moving to the withdraw point.

Vehicles can use "suppression fire" while withdrawing; that is, a vehicle "remembers" an enemy location for 30 seconds after the enemy disappears. If the disappearing target is moving, the withdrawing vehicle calculates the anticipated target location before firing.

All vehicles perform an Occupy Position task when reaching the withdraw point. If the enemy is detected, the area to occupy is in the direction of the enemy's center of mass. If the enemy is not detected, the area to occupy is in the direction of the initial withdraw position. The vehicles occupy the position until you assign another task.

4.3.4.22 Follow Unit

The Follow Unit task runs on the trailing unit. Each of the vehicles in the trailing unit follows a vehicle in the leading unit. DI/IFV platoons are the main users of this task. In this capacity, this task is called from the Unit Mixed Travel task. The Unit Mixed Travel task passes the leading unit and following unit to the Unit Follow task. The Unit Follow task spawns individual movement tasks on the entities in the following unit with the entity to follow.

A pairing between IFV and DI exists in the mounting and dismounting tasks. As a result, each individual DI follows the same IFV that it mounts or each individual IFV follows the same DI that mounts it.

4.3.4.23 Breach

The unit-level Breach task divides the unit into two functional groups. The first group (the occupy group) performs an Occupy Position; the second group (travel) travels through the area. When the travel group reaches the end, it occupies position, and the group that was performing an occupy position travels along the same route.

The vehicles in the unit are divided (as much as possible) between the two groups. If there is only one vehicle in the unit, it is assigned to the travel functional group, which always contains at least one vehicle. If there is an odd number of vehicles (other than one), the extra vehicle is assigned to the Occupy Functional group.When mines are encountered they explode but do no damage to the vehicles during the breach. The vehicles move at 7 kph in a close staggered-column formation along a given route. The Mixed Breach task mounts the DI (if they exist) before spawning the Breach task. You can set the React to Indirect Fire task to spawn a Breach task when it detects exploding mines. (Alternately, the React to Indirect Fire task can react to mines by spawning a Mine Withdraw task.)

4.3.4.24 Mine Withdraw

When a unit-level enters a minefield, the unit Indirect Fire Reaction task spawns the Mine Withdraw task. The Mine Withdraw task spawns the Backtrack task on the subordinates causing the vehicles to backtrack for a given distance. A unit Withdraw task is then executed.

4.3.4.25 Attack by Fire

The unit-level Attack by Fire task performs an Occupy Position, and fires at the enemy using the alternating fires technique. A call for indirect fire is reported by radio at the beginning of the attack, and a spot report is sent when the attack is over.

4.3.4.26 React to Air

React to Air Attack is a unit-level reaction handling task that causes platoon vehicles to respond to an air attack by scattering. The platoon scatters at high speed when it sees enemy aircraft or receives an impact packet from an air vehicle. If the platoon is in a defensive position, no scattering takes place.

4.3.4.27 React to Indirect Fire

React to Indirect Fire is a unit-level reactive task that responds to indirect fire in the immediate area (within 50 meters from a vehicle in the platoon). This task causes the platoon to speed up when receiving an artillery impact packet. After a certain time of receiving no indirect fire, the platoon returns to its original pace. If the platoon is stopped, then nothing happens in response to indirect fire. This task also monitors minefield explosions and executes the appropriate reaction (see Mine Withdraw).

4.3.4.28 Supply

The Unit Supply task implements resupply of a specified unit. The supply vehicle iteratively moves from one vehicle to the next within a unit and resupplies the individual vehicles.

The Resupply task transfers munitions and fuel from resupply vehicles such as the US M977 and M978, and USSR URAL375C and URAL375F, to artibrary combat vehicles. The Resupply task uses both the Distributed Interactive Simulation (DIS) protocol 2.0.3, and the Standard SIMNET protocol to communicate transfer information.

4.3.4.29 MLRS

The MLRS task implements the movement characteristics of a Multiple Launch Rocket Unit performing an indirect fire mission. These movements include moving into a user-specified firing position, and moving into a user-supplied hiding position upon completion of firing.

4.3.5 Mixed Platoon Tasks

Mixed Platoon-level tasks are for use by DI/IFV platoons or any mixed platoon.

When you assign a taskframe at the mixed platoon level, the mixed level task spawns the necessary tasks on the subordinate platoons or subgroups. When a frame is assigned at the platoon level rather than at the mixed platoon level, the frame spawns the necessary unit-level tasks to the assigned platoon only.For example, if a Move frame is assigned on a platoon of M1s, it spawns itself a Traveling task. However, if the Move frame is assigned on an IFV/DI platoon and the DI are in a dismounted state, the Mixed Move task spawns a movement frame on each of the two subordinate platoons. The DI platoon gets a frame that includes a Traveling task; the IFV platoon gets a frame

that includes a Follow Unit task with the following offsets set so that a DI follows in front of an IFV. This enables the DI and IFV to move in a coordinated pattern.

Tasks for mixed platoon ground units are described as follows:

4.3.5.1 Mixed Prep Occupy Position

This task effectively runs a unit-level preparatory task for the Occupy Position task on mixed platoons consisting of more than one type of entity. It serves as a pass-through task for the normal platoon-level Prep Occupy Position task.

4.3.5.2 Mixed Targeter

This mixed platoon task spawns a unit-level Targeter task on each of its subordinate units. It is a pass-through task for the normal platoon-level Targeter task.

4.3.5.3 Mixed Overwatch Move

The Mixed Overwatch Move task determines if the unit is a mixed platoon or a normal platoon. Depending on the platoon type, the task either spawns a task that implements overwatch behavior, or spawns pass-through tasks that implement overwatch behavior.

If the unit is a normal platoon, the Mixed Overwatch Move task spawns itself a Unit Bounding Overwatch task. If the unit is a mixed platoon, the Mixed Overwatch Move task spawns tasks appropriate to the subordinate units based on whether those units have mounted or dismounted DIs.

For a mixed platoon with dismounted DIs, the Mixed Overwatch Move task spawns an Overwatch Move task on the subordinate DI platoon. The route is the unit route. It then spawns a Follow task on the subordinate IFV platoon with the DI platoon leader as the lead vehicle.

For a mixed platoon whose DI are mounted, the Mixed Overwatch Move task spawns a unit-level Overwatch Move task on the subordinate IFV platoon.

4.3.5.4 Mixed Traveling

The Mixed Traveling task implements a mixed platoon-level Movement task. This task determines whether the unit is a normal platoon or a mixed platoon. If the unit is a normal platoon, the Mixed Traveling task spawns itself a unit-level Traveling task.

When the unit is a mixed platoon, the Mixed Traveling task spawns the appropriate tasks on the subordinate units. For a mixed platoon whose DI are mounted, the Mixed Traveling task spawns a unit-level Traveling task on the subordinate IFV platoon.When the unit is a mixed platoon whose DI are dismounted, the Mixed Traveling task examines the Y offset. If the Y offset (following) parameter is negative, the IFVs must follow the DI. If the Y offset parameter is positive, the DI must follow the IFVs. Therefore, the Mixed Traveling task spawns a unit-level Traveling task on the subordinate leading unit (either the DI or IFV platoon) with the route as the unit route. It then spawns a Unit Follow task on the subordinate following unit with the leading unit as the lead unit.If the mixed platoon contains a tank, the tank leads and the subordinate platoon that is issued a unit-level Traveling task follows the tank.

4.3.5.5 Mixed Halt

The Mixed Halt task can run a unit-level Halt task on mixed platoons consisting of more than one type of entity. The Mixed Halt task is a pass-through task for the normal unit Halt task.

4.3.5.6 Mixed Withdraw

The Mixed Withdraw task can run a unit-level Withdraw task on mixed platoons consisting of more than one type of entity. The Mixed Withdraw task is a pass-through task for the normal unit Withdraw task.

4.3.5.7 Mixed Concealment

The Mixed Concealment task can run a Unit Concealment task on mixed platoons consisting of more than one type of entity. The Mixed Concealment task is a pass-through task for the normal unit Concealment task.

4.3.5.8 Mixed Delay

The Mixed Delay task can run a Unit Delay task on mixed platoons consisting of more than one type of entity. The Mixed Delay task is a pass-through task for the normal unit-level Delay task.

4.3.5.9 Mixed Breach

The Mixed Breach task provides the capability to run a unit-level Breach task on mixed platoons consisting of more than one type of entity. It is a pass-through task for the normal unit-level Breach task.

4.3.5.10 Mixed Mine Withdraw

The Mixed Mine Withdraw task can run a unit-level Mine Withdraw task on mixed platoons consisting of more than one type of entity. The Mixed Mine Withdraw task is a pass-through task for the normal unit-level Mine Withdraw task.

4.3.5.11 Mixed Dismount

The Mixed Dismount task dismounts a group of subordinate DI from its corresponding IFVs. The Mixed Dismount can be issued at the mixed level or the individual DI or IFV level.

At the mixed level, the initial action is to issue a unit-level Halt task to the subordinate DI and IFV platoons. After the subordinate platoons are halted, the Mixed Dismount spawns a unit-level Dismount task on the IFV platoon. The unit Dismount task dismounts all the possible pairs of IFV and DI using the task organization index, and waits for all subordinate IFVs to complete the dismount before ending.

When the Mixed Dismount task is spawned on an individual IFV or DI vehicle, it halts the corresponding IFV or DI. Once halted, the Mixed Dismount spawns a Dismount task on the IFV vehicle.

4.3.5.12 Mixed Mount

The Mixed Mount task mounts a group of subordinate DI onto its corresponding IFVs. The Mixed Mount can be issued at the mixed level or the individual DI or IFV level.

At the mixed level, the initial action is to issue a Halt task to the subordinate DI and IFV platoon. After the subordinate platoons are halted, the Mixed Mount spawns a unit Pick Up task on the IFV platoon. The unit Pick Up task mounts all the possible pairs of IFV and DI using the task organization index, and waits for subordinate IFVs to complete the pickup before ending.

When the Mixed Mount task is spawned on an individual IFV or DI vehicle, it halts the corresponding IFV or DI. Once halted, the Mixed Mount spawns a unit Pick Up task on the IFV vehicle.

4.3.6 Company Tasks

ModSAF has several company-level tasks. These tasks are for use by M1, M2, M1A2, M2-reinforced, T72M, T80, BMP1, BMP2, and BMP2-reinforced companies.

When a taskframe is assigned at the company level, the company task spawns necessary tasks on its subordinate units. For example, if a movement frame is assigned on an M1 company, the company task spawns a movement frame on each of its subordinate platoon and vehicle units.

4.3.6.1 Company Halt

The Company Halt task halts an entire company by spawning unit-level Mixed Halt tasks on every subordinate (platoons and vehicles). The vehicles stop and wait for orders. This task is also the preparatory task for other company-level tasks such as Company March.

4.3.6.2 Company March

The Company March task moves an entire company on either a cross country route or a road route. It functionally organizes the extra vehicles (CC, XO, and others) into the platoons and then spawns Mixed Traveling tasks on the platoons. The platoons remain in formation by having a subordinate follow an assigned lead subordinate with an offset based on the formation. The

company travels to the end of the route at which point the task removes the lead subordinate assignation and converts the task organization back to the original organization.

4.3.6.3 Company React to Air

The company-level React to Air task is a reaction handling task that causes company reaction to air vehicles. The platoons in the company scatter when they detect enemy aircraft or receive an impact packet from an air vehicle. However, they scatter only if the vehicles are not in a defensive position. This task is similar to the unit-level React to Air task.

4.3.6.4 Company Occupy/Defend

The Company Occupy/Defend task implements a company-level version of the unit-level Prep Occupy Position task. You specify a battle position, an optional left TRP, an optional right TRP, and an engagement area TRP. The Company Occupy/Defend task divides the battle position into "n" segments of equal length, "n" being the number of platoons in the company.A separate PO line object, as well as left and right TRPs, are created for each of the platoons in the company. The company commander and executive officer are functionally organized into one platoon each. Unit-level Occupy Position tasks are then spawned for each of the platoons.

4.3.6.5 Company Withdraw

The Company Withdraw task enables companies to withdraw to a particular point. The task creates final points for each platoon to withdraw to based on the user-specified point and a parametric data-specified width. The platoon end points are placed on a line that is perpendicular to the line formed by the company position and the point you specified.

This task functionally organizes the command vehicles into correct platoons. It divides the platoons into two groups: an occupy group and a withdraw group. The occupy group occupies position while the withdraw group withdraws to a certain position. The withdraw task performs an occupy position when the vehicles reach the withdraw point. The groups switch functions when the withdraw group occupies position. This process continues until all groups reach their final withdraw point. The platoons occupy position at their final withdraw points.

When there are three platoons in the company, the center platoon is the occupy group and the remaining two platoons comprise the withdraw group. When there are two platoons in the

company, one platoon is the occupy group and the other is the withdraw group. When there is one platoon in the company, that platoon is the withdraw group, and it withdraws directly to the final point.

4.3.6.6 Company Attack

The Company Attack task implements a basic attack for a company. The platoons march in a line formation toward the attack objective. When the company reaches the attack objective (a parameter), it occupies a battle position facing the direction of the attack. If the number of casualties is too high, the company stops short of the objective and occupies a position. This task is similar to the unit-level Assault task.

4.3.6.7 Company Targeter

The Company Targeter task spawns Mixed Targeter tasks on all of the subordinates, and is necessary to the Company Attack task. It is similar to the Company Halt task.

4.3.6.8 Company Occupy Assembly Area

The Company Occupy Assembly Area task places platoons in positions to provide 360 degree coverage. Battle positions in a quantity "n" and the associated TRPs are created for the Mixed Occupy Positions tasks, where "n" is the number of subordinate units comprising the company.

4.3.7 FWA Unit Tasks

FWA fly in a tactically correct formation along straight route segments, in shallow turns, in hard turns, in orbit, and while changing formations. They perform tactically correct attacks on planned targets and on targets of opportunity for all legal combinations of attack geometry, entry, and delivery. Tasks for FWA units are described as follows.

4.3.7.1 Flight FWA Ground Attack

The unit-level Flight FWA Ground Attack task controls the phases of the attack for vehicles in the flight unit. It coordinates FWA Ground Attacks that can consist of flights of 1 to 4 aircraft.

This task uses the following user-specified parameters: target location (which is a point or text graphic), attack route (optional), speed, altitude, formation, attack geometry, attack entry, attack delivery, and target priorities list. During an attack, the target priorities specified in this task override the target priorities that are specified in the global Rules of Engagement for the unit. They are restored to their previous global Rules of Engagement value when the attack ends.

When a flight is directed to execute a split or 90/10 attack, that flight must be divided into two sections or functional subgroups. In non-split attacks, the FWA ground attack task assigns itself a Section ground attack frame. This is analogous to the Mixed Travel task which either distributes unit tasks to subordinate DI and vehicle sections, or assigns itself a Unit Travel task.

4.3.7.2 Section FWA Ground Attack

The unit-level Section FWA Ground Attack task spawns a Follow Route task on the leader, and spawns FWA Formation Keeping tasks on the followers to get the aircraft to the Ingress Point (IP).

When the unit reaches the Ingress Point, FWA Ground Attack tasks are spawned on all vehicles with the appropriate parameters for their role in the attack, and the attack begins. During the attack, when all attack tasks are complete, the rendezvous location is computed before the task transitions to the rendezvous state. In the rendezvous state, when all vehicles have reached the rendezvous location, the task ends.

4.3.7.3 Targets of Opportunity

The Targets of Opportunity task is a reactive task that searches for Targets of Opportunity and triggers an attack if any are detected.

This task uses the following user-specified parameters: do/do not attack targets of opportunity boolean, maximum distance off the unit's route to pursue a target of opportunity, maximum angle off the unit's direction to pursue a target of opportunity, target priority list for targets of opportunity, and the parameters required for the Flight FWA Ground Attack task as specified previously (i.e. speed, altitude, formation, attack geometry, attack entry, attack delivery, and target priorities list).The Targets of Opportunity task queries the Unit Enemy task for any live enemy vehicles the unit may have detected. It processes each enemy vehicle through the filter defined by the distance, angle, and target priority parameters. When an enemy passes the filter, the Attack Ground Targets taskframe executes.

4.3.8 RWA Unit Tasks

Tasks for RWA units are described as follows.

4.3.8.1 RWA Unit Hover

RWA Unit Hover is a unit-level task which enables RWAs to hover in their current positions at the specified altitude. RWA Unit Hover is also used as a preparatory task in which case the specified altitude is ignored, and the positions of the vehicles in the unit do not change even if the vehicles are currently on the ground.

4.3.8.2 RWA Unit Land

RWA Unit Land is a unit-level task which enables RWAs to land in their current positions. If the vehicles are currently on the ground, they remain as is.

4.3.8.3 RWA Unit Orbit

RWA Unit Orbit is a unit-level task that spawns the Fly Route tasks on each of the subordinates to enable them to follow a semicircle route. Once a unit has finished the route, the remainder of the circle is given as a route. The unit circles in RWA-trail formation continuing until an On Order is received.

4.3.8.4 RWA Unit Assemble

The RWA Unit Assemble unit-level task halts RWA subordinates in a coil formation. If the vehicles are currently on the ground, they ascend in order to land and assume a coil formation on the ground.

4.3.8.5 RWA Unit Attack

RWA Unit Attack implements an attack for rotary wing aircraft. Two types of attack can be specified: Hover and Running Fire.The Hover attack spawns the Fly Route task to move the RWA

unit into the area of the objective. Once the the unit is moved, an Occupy Position task is spawned with the battle position in the shape of a "V." The RWA vehicles move to the positions and then execute an RWA Unit PopUp task. The RWA unit remains and performs a PopUp attack until an On Order occurs.The Running Fire Attack task spawns a unit-level Running Fire Attack task on the unit. Refer to the next section.

4.3.8.6 RWA Unit Running Fire

RWA Unit Running Fire is a unit-level task that causes the aircraft to ascend and fly toward a user-specified attack objective while searching for targets. Once targets are detected, the vehicles move toward the attack objective while shooting at the enemy vehicles.

Vehicles turn around and fly toward their starting positions if one or more of the following conditions exist:

- Vehicle gets too close to the attack objective.
- Vehicle destroys its intended target.
- Vehicle runs out of ammunition.

Once the vehicles reach their starting point, they attack again if they have ammunition and targets still exist around the attack objective.

4.3.8.7 RWA Unit PopUp

RWA Unit PopUp is a unit-level task that spawns vehicle-level PopUp Attack tasks on each of the subordinates. The task cycles through the subordinates and assigns the vehicle PopUps individually, allowing only one RWA in the unit to "pop up" at a time. This task is not accessible from the user interface, but can be accessed through the RWA Unit React Contact and RWA Unit Attack tasks.

4.3.8.8 RWA Unit Prep Occupy Position

The RWA Unit Prep Occupy Position task implements a unit-level Preparatory Occupy Position task similar to the Prep Occupy Position task supported for ground vehicles.This preparatory task finds covered and/or concealed positions along the battle position and instructs the subordinates

to move toward these positions. Based on the battle position and the number of subordinates, both the number of vehicles per segment and the battle areas are calculated.

The subordinates are ordered based on geometry, and are assigned positions across the battle position ensuring that no vehicle crossover occurs while they are traveling to their positions. Once they reach the positions, the aircraft ascend, if necessary, using the Fly Route task to travel the route. The vehicles either land if the specified end altitude is 0.0, or they hover at the specified end altitude.

Note: The RWA Unit Prep Occupy Position task is available in both the preparatory and actual phases of the execution matrix menu. However, RWA Unit React to Contact is the actual task used in that taskframe.

4.3.8.9 RWA Unit React to Contact

cindex RWA Unit React to Contact The RWA Unit React to Contact task monitors enemy vehicles, and implements a basic reaction for RWAs that is used in the actual Occupy Position taskframe.

If enemy vehicles are detected, the RWA Unit PopUp task is spawned on the unit. The PopUp task continues until the enemy vehicles are no longer present. The RWA Unit React Contact task then stops the PopUp and returns to monitoring for enemy vehicles. This task can be overriden by the stop reaction button on the user interface. *Note:* This task is used is for the Occupy Battle Position taskframe only.

4.3.9 Unit Taskframes

The ModSAF commander creates missions for the units to control by combining and editing sequences of taskframes. Unit taskframes are created from unit tasks.

This section lists the unit taskframes defined in the ModSAF system.

Some taskframes do not apply to all units. The specific taskframes that can be executed are determined by the unit to which they are assigned.

The tasks that compose a taskframe have many parameters with default settings. You customize the taskframes by editing these parameters that include speeds, formations, routes, and so forth.

4.3.9.1 Ground Unit Taskframes

Ground unit taskframes include:

Assault — Inputs an object (expressed as a line, point, an area, or text), an optional route, and an optional speed. The unit charges the enemy. If the assault is part of a reaction, the unit continues its previous mission once the enemy is defeated. If the assault is an assigned mission, the unit occupies a battle position.

Halt — Stops unit movement. (A unit performing a road march is positioned in a herringbone formation.)

Occupy Battle Position — Attempts to find covered positions. Takes as its input a multi-segment line and three TRP points. Two optional points define the sector for fire and a third defines an engagement area. The frame places the vehicles somewhat equally along the battle position line. In addition, the frame searches for hull down positions for the vehicles enabling them to have line of sight to the engagement area.

Move — Performs a tactical march.

Roadmarch — Performs a road march using the same tasks as the Move frame.

Withdraw — Moves a group of vehicles away from the enemy, and performs an Occupy Position until another order is given to the unit. Armored vehicles withdraw in reverse gear until the enemy is no longer visible.

Assemble — Stops unit movement and places a unit in coil formation.

Mount — Mounts infantry in mixed units.

Dismount — Dismounts infantry in mixed units.

Concealment — Attempts to find concealed positions.

Breach — Moves a group of vehicles slowly through a minefield area. The mines explode if encountered, but do no damage to the vehicles during the breach. The vehicles are divided into two functional groups: the first group implements an Occupy Position task; the second group travels through the area. When the travel group reaches the end, it implements an Occupy Position task. The first group then travels along the same route.

Supply — Implements resupply of a vehicle. The task processes the necessary issue and reception of supply protocol packets.

Delay — Runs a Unit Delay task on mixed platoons consisting of more than one type of entity. The Mixed Delay task is a pass-through task for the normal unit-level Delay task.

MLRS Implements the firing and mission acknowledgement characteristics of Multiple Launch Rocket vehicles. It loads the ammunition included on the vehicle, waits until the vehicle is in firing position, and then fires.

Attack by Fire

Performs an Occupy Position; fires at the enemy using the alternating fires technique. A call for indirect fires is reported by radio at the beginning of the attack, and a spot report is sent when the attack is over.

Overwatch Movement

Performs Bounding Overwatch movement in which a portion of the unit, using terrain features as cover, covers the movement of another portion of the unit.

Pursue Lets a unit "chase" another unit.

Follow a Vehicle

Enables vehicles in a trailing unit to follow vehicles in the leading unit.

4.3.9.2 Rotary Wing Aircraft Taskframes

The ModSAF software can perform the following taskframes for RWA units:

Fly Route RWA pairs and platoons move on routes in contour and low-level flight as well as line, wedge, echelon-left, echelon-right, trail, staggered-left, and staggered-right formations.

Hover Frame

Implements an RWA hover. The hover altitude and direction can be specified. The RWA stops, reaches the specified altitude above ground level, and faces the specified direction.

Land Frame

Lands RWA by flying the vehicle to a landing spot. The RWA stops, faces a given direction, and descends until it lands.

Assemble Frame

Halts a group of RWA subordinates and places them in a coil formation. If the vehicles are landed, they take off in order to land and assume a coil formation.

Attack Frame

Implements an RWA attack. Two types of attack tasks can be specified: Hover Attack and Running Fire Attack.

The Hover Attack task spawns the Fly Route task that moves an RWA unit to the area of the objective. Once the the unit is there, an Occupy Position task is spawned with the battle position in the shape of a "V." The RWAs move to the positions and then

execute an RWA Unit PopUp task. The RWA unit performs a PopUp attack until an On Order occurs.

The Running Fire Attack task spawns a unit-level Running Fire Attack task on the unit.

Occupy Position Frame

Implements a unit-level preparatory task for RWA Occupy Position similar to the Prep Occupy Position task supported for ground vehicles. It finds covered and/or concealed positions along the battle position and instructs the subordinates to go toward these positions. Once the positions are found, the vehicles ascend, if necessary, using the Fly Route task to travel the route. The vehicles land if the specified end altitude is 0.0, or they hover at the specified end altitude.

4.3.9.3 Fixed Wing Aircraft Taskframes

The ModSAF software can perform the following FWA taskframes (missions):

Combat Air Patrol (CAP)

When an FWA is patrolling an area or waiting for orders or targets, the CAP taskframe implements one-to-one air-to-air combat. Aircraft executing this frame do not attack ground targets.

Return To Base

FWAs fly to a point designated as their base of operations. Upon reaching this point, the FWA land, refuel, and rearm.

Sweep FWAs fly to a point or along a route. Upon reaching the destination point or end of the route, the vehicles orbit. Note that the Sweep taskframe performs one-to-one air-to-air combat. Aircraft executing this frame do not attack ground targets.

Ingress Assigns a route to the target position. This task contains the FWA Fly Route task that controls the unit's movement along the ingress route. It uses the Targets of Opportunity task to attack targets of opportunity that the unit encounters.

Attack Ground Targets

Issues an FWA unit attack on a target position. This taskframe contains the FWA Ground Attack task that controls the unit's movement during the attack. It uses the Targets of Opportunity task to attack targets of opportunity that the unit encounters during ingress and egress phases.

Egress Assigns a route for FWA to follow after attacking the target. It contains the Follow Route task that controls the unit's movement on the egress route, and uses the Targets of Opportunity task to attack targets of opportunity that the unit encounters.

4.3.10 Unit Reactions

A Unit Reaction refers to changes in the battlefield environment. For example, a platoon of tanks on a road march may execute a new taskframe in response to spotting enemy. This new frame may issue a different set of tasks to execute. (The parameters of these tasks can be customized to the specific environment at runtime.)

Different ModSAF units can have unique sets of reactive behaviors that handle different situations and battlefield events. Reactions include, but are not limited to, the following behaviors.

4.3.10.1 Ground Unit Reactions

The ground unit reactions include the following:

- Contact Drill - Triggered by contact with enemy units of appropriate size. The unit continues on its mission.
- Attack by Fire - Causes a unit to attack from a battle position.
- Assault - Causes a unit to attack from an on line position.

4.3.10.2 Aircraft Reactions

An aircraft checks for bingo fuel by monitoring its fuel load and its distance from its base or refuel location. If the critical fuel level is reached, the vehicle flies to the base or refuel point and then lands.

4.4 Parser Interface

The SAFsim has a parser interface with the following capabilities for testing ModSAF software:

- A limited set of command line instructions for controlling the ModSAF system and vehicle debugging.
- The capability to turn debugging code on and off in the SAFsim.
- The capability to query the status of executing tasks and units in the simulation exercise.

Enhancements to the parser include:

- Persistent Delete All. Deletes all entitites and overlay features in the PO database. Serves as a refresh function.
- Persistent Release All. Issues an On Order authorization to units with pending On Order authorizations.
- sys dump (file name). Performs a window dump of the PVD to the named file.
- Command Line Arguments. A switch has been added to the command line that directs the parser to load the source file for the parser.

5 ModSAF Logger

The ModSAF data logger, or SAF-logger, lets you record and play back simulation exercises. It records PO protocol packets to allow initialization of ModSAF from any point in a logged exercise.

The ModSAF Logger has a graphical user interface, and provides access to all features that the data logger supports.

Note that this scheme works on the analogy of a CD player. Just above the "Loop Play Controls" section, a new section has been added called "Event Flag Controls." It has a series of buttons similar to the buttons at the top of the interface, including:

```
 -----   -----   -----   -----------------------
|  <  | |  >  | |  0  | |(same as "Loop" button) |
 -----   -----   -----   -----------------------
 Last    Next    Scan    Preview
```

Note:

- "Last" rewinds the logger to the previous time-ordered event flag in the log file.
- "Next" advances the logger to the next time-ordered event flag.
- "Scan" jumps to the next time-ordered event flag, plays for 21 seconds seconds, continues to the next event flag, and repeats these actions.
- "Preview" loops on the current event flag.

Directly underneath these buttons are a series of windows identifying the "Index," "Category," and "Memo" of the current event flag. The windows should look something like the following:

```
         ---              ----------        ----------
 Index  | 0 |   Category | Bookmark |  Memo |          |
         ---              ----------        ----------
```

Note:

- "Index" is the number of the current event flag.
- "Category" identifies the category of the event.
- "Comment" event is generated in the voice logger when the operator presses the comment button on the microphone stand.

- "Bookmark" event is generated when the operator presses the "Add Bookmark Event" button (see below).
- "Memo" provides a space for the operater to add a textual comment to the event.

Beneath the editable windows is a series of buttons:

```
 ------------       --------------    --------------------
| Go To Event |    | Delete Event |  | Add Bookmark Event|
 ------------       --------------    --------------------
```

Note:

- "Add Bookmark Event" flags a "bookmark" category event at the current log time.
- "Delete Event" deletes the current event flag.
- "Event List" pops up the list of events, displaying the Time, Index, Category and Memo fields. You can select any event from the list in the same manner as selecting a file, and move to that point in the log file.

Events are written to the data file, and can be modified as necessary.

5.1 Exercise Recording

The ModSAF data logger records the simulation packets of any protocol family transmitted on the simulation network. These include the DIS, SIMNET, Persistent Object (PO), Radio, Radio Signal, and Data Collection protocols. The GUI lets you choose which protocol families to record. It also lets you specify the exercise ID, exercise start time and date, and the file name under which the exercise is to be saved. A compact data storage format enables large-scale exercises that last many hours to be recorded into a single file. The data is recorded in a way that permits random access into the recorded exercise.

The ModSAF data logger supports a studio mode with which it can record to multiple logger files simultaneously, playback multiple logger files simultaneously, and edit and splice logger files.

The ModSAF data logger has an automatic shut-off feature for terminating the recording or playback at a user-specified time. This feature has the same effect as pressing the "stop" button at the specified time.

5.2 Exercise Playback

The ModSAF data logger can play back the packets of any protocol family recorded in a ModSAF data logger file. The GUI allows you to select which protocol families to play back.

The ModSAF data logger displays exercise statistics on the GUI in real time while an exercise is being played back. These statistics include the exercise packet rate, exercise entity count, logger entity tick rate, elapsed exercise time, and remaining exercise time.

The ModSAF data logger can play back a data logger file on any exercise ID, regardless of the exercise ID on which the file was recorded. The ModSAF data logger supports modification of entity simulation IDs, so that multiple exercises can be played back simultaneously without interference.

The ModSAF data logger can play an exercise back in forward and reverse directions. It can play back either in real time, up to 50 times faster than real time, or up to 10 times slower than real time. It compensates for first order dead reckoning of entities so that they appear to move smoothly even when being played back at speeds other than real time. The logger can play back a user-specified portion of an exercise repeatedly (loop play) and it can pause an exercise playback without causing the entities to time out on the simulation network.

5.3 Scenarios from a Logged Exercise

Since all ModSAF command and control information is sent through the PO protocol, the ModSAF data logger can record the state of the missions that ModSAF entities are executing throughout an exercise. This recording does not interfere with the normal operation of the simulation.

The logger GUI provides an interface for generating a ModSAF scenario file at any point during exercise playback. A scenario file generated from a logged exercise can be used to initialize ModSAF entities from that point in the exercise. This interface allows ModSAF entities to be initialized from a chosen point in the exercise without having to replay the entire exercise up to that point.

6 ModSAF Interfaces

ModSAF components interface by way of the following databases:

- Simulation database
- PO database
- Parameter database
- Terrain database

6.1 Simulation Database Interface

The Simulation Database contains information about the physical state of the battlefield and the entities in it. This information includes entity state as well as impact, collision, and fire events. Access to the entity information is obtained from the entity ID or the entity's geographic location.

Entities in the database are simulated either locally or remotely. Since a SAFstation does not perform simulation, all its entities are remotely simulated.

The Simulation Database maintains a seamless interface to both local and remote entities. Lazy evaluation of dead reckoning is performed by the database. Events are queued for the local entities to whom they are of interest.

The ModSAF software supports the DIS 1.0 and the DIS 2.0.3 protocols, with appropriate extensions necessary for ModSAF, such as radar packets. All applicable simulation packets (entity state, events, exercise control, appearance, impact, status, etc.) are supported. In accordance with the DIS standard, each entity broadcasts its state at least once every five seconds, and more often if required by dead reckoning algorithms. The DIS protocol uses the UDP/IP network interface layers.

The ModSAF software also supports the SIMNET 6.6.1 protocol using the SIMNET association layer as the network interface layer.

The network drivers are in a small, well-defined interface module to enhance portability across operating systems and computers.

6.2 PO Database Interface

The Persistent Object (PO) database supports representation of command or exercise information required to set up exercises and control their participants. This information includes unit and entity locations, unit and entity missions, and mission-specific information (including graphics and routes). Sharing of this information is possible among workstations representing the same side in an exercise.

The PO database supports large numbers of hosts and real-time performance. It can handle both query-driven and event-driven interfaces. It supports thousands of database objects that change infrequently (less than two or three times per minute). Objects rebroadcast with a 30-second timeout period for an initial period. Source-based filtering is used to reduce steady-state packet rates.

The PO database can recover from missed packets and can handle migration of simulation objects from one simulation computer to another. Simulation objects can migrate automatically during simulation and entity creation, either for load balancing or graceful recovery when a simulator times out.

6.3 Parameter Database Interface

The parameters and models used in the construction of the ModSAF system are contained in a set of modular knowledge bases or parameter files, The parameters defined in these public parameter files initiate a runtime parameter database. Since this database can be changed, it permits the modification of models without the recompilation of source code.

The types of parameters defined in the public files are described below.

6.3.1 Organizational Parameters

Organizational parameters define the organic echelons and formations used by the ModSAF units. These echelons can be grouped to define new unit types.

6.3.2 Entity Parameters

The entity parameters specify characteristics of ModSAF entities and define the component

physical models and weapons systems for each ModSAF entity. The network entity types are defined in the simulation protocol files, but variations of these entities can be created in the parameter files. The following entity information can be specified in these files:

- Network representation
- Alignment, identity, and function of entity
- Types of weapons, including bombs, missiles, guns
- Weapon system characteristics including: range, round velocity, mass, guidance model, maximum speed
- Dynamics model parameters
- Damage models for direct and indirect weapons
- Standard fuel and ammunition loads

6.3.3 Behavioral Parameters

The behavioral parameter files define parameters for the various behavioral tasks, specializing them for different units and situations. Tasks and their arameters are also organized into task frames to define tactics and mission components.

6.3.4 User Interface Parameters

The user interface parameter files define parameters for the user interface. These parameters can specify: unit icons and entity pictures; graphics attributes (color, line types); size of panes for map, messages, and editor; control measures; and coordinate conversion.

Some user interface parameter files define help messages that appear on the GUI. Others are editor definitions of the pieces of data that the user can specify from the various GUI editors. Editor parameter files indicate the editor title and, for each editor field, specify field name, type, optional or required status, and initial value.

6.4 Terrain Database Interface

A ModSAF terrain database consists of two separate representations of the terrain, each managed by its own ModSAF library.

- The polygonal database, CTDB (Compact Terrain Database), contains elevation soil type, and feature data. It is managed by libctdb. *Note:* As of V1.5.1, CTDB formats are revised as follows:
    1. Format 1 is the first CTDB format.
    2. Format 2 contains new types of microterrain to support some database features in a non-S1000 derived source format. In format 1, there is only one kind of microterrain. In format 2, two non-padding bits which were guaranteed to contain zeros in format 1 are reused to distinguish four types of microterrain. Thus, a format 2 database cannot be used with a format 1 version of libctdb. However, a format 1 data can be used with a format 2 version of libctdb.
    3. Format 3 adds topology and abstract features. These features do not exist in the format 1 or 2 versions. The format 3 version of libctdb can process format 1 and 2 databases. However, applications using format 3 may not work properly since these features are not present in older databases.

    Conversion between formats can be done using the '`recompile`' program.
- The quadtree representation, which is object oriented and contains features and attribute information (including road and river networks). It is managed by libquad (library for quadtree which indicates how the objects are spatially organized for efficient searching).

A function that needs to access the terrain database uses the representation and library that is most efficient. For example, the plan view display, which displays a two-dimensional representation of the terrain, uses the polygonal database for intervisibility displays and the object-oriented database for feature drawing.

Each ModSAF component has its own copy of the terrain database. This allows components to communicate with pointers into data structures so that terrain information does not need to be sent over the network.

CTDB functions are used to perform the following operations:

- Display contour lines, hypsometric tinting, and terrain cross section.
- Calculate intervisibility (including terrain and vehicle blockage), for detection and targeting.
- Generate a rotation matrix to place vehicles on the terrain, both for elevation and orientation.
- Look up elevation lookup along a line segment (find high ground, find terrain profile, etc.).
- Calculate slope and obtain soil type, which is used by the vehicle dynamics model to set vehicle maximum speeds. The terrain database includes the soil types (road, muck, deep water, shallow water, packed dirt, soft dirt, sand, forested, etc.) at each point.
- Flyout projectiles (and air vehicles) to detect ground collisions.

- Calculate radar clutter.
- Convert coordinates to a requested coordinate system.

Quadtree functions are used to perform the following operations:

- Draw the two dimensional SAFstation map display.
- Perform route generation to create road routes, using the road network, and to check for water crossings on all ground routes, using the road and river networks. It can also generate routes around forested areas (tree canopies) and lakes.
- Perform local terrain reasoning for each simulated entity by keeping track of its adjacent terrain features. These features are used as input for controlling movement so that water is avoided during cross country travel, and so that obstacles, like buildings, are avoided. Area features, like tree canopies or boulder fields, are also avoided.
- Identify terrain objects for use for cover and concealment during missions and reactions.

Appendix A References

This appendix describes and references ModSAF papers that were presented at the *Fourth Conference on Computer Generated Forces and Behavioral Representation*, May 1994.

The papers include:

1. "Operator Control of Behavior in ModSAF" by Andy Ceranowicz, Dan Coffin, Joshua Smith, Roger Gonzalez and Carol Ladd.
2. "Near-term Movement in ModSAF" by Joshua E. Smith.
3. "The Incorporation of Validated AMSAA Combat Models into ModSAF" by Anthony J. Courtemanche and Paul Monday.
4. "Cover and Concealment in ModSAF" by Michael Longtin.
5. "Benchmarking and Optimization of ModSAF 1.0" by Wendy Richardson and G. Robert Vrablik.

A.1 Operator Control

Control of the behavior of the simulated entities is shared between the commander and the ModSAF system. The commander provides supervisory control over the forces and intervenes when the automated decision logic is not sophisticated enough to handle a situation satisfactorily. The commander can control behavior in three ways: creating preplanned missions, setting up automated reactions, and issuing immediate commands. These control methods are described in the paper, "Operator Control of Behavior in ModSAF" by Andy Ceranowicz, Dan Coffin, Joshua Smith, Roger Gonzalez and Carol Ladd.

To express a preplanned mission, ModSAF uses an execution matrix augmented by control measures drawn on the map display. The execution matrix, a standard Army tool for planning and for command and control, allows an officer to sequence and synchronize the actions of his units. The matrix divides the mission into phases and indicates what each unit should be doing in each phase. The graphical control measures are arguments to the mission. This approach makes it possible to define and sequence the mission phases.

Reactions are implemented using the ModSAF task architecture with each reactive trigger represented by a task. (An example of a reactive trigger is the Actions On Contact task.)

A reactive trigger task remains running even after it invokes a reaction. This approach supports behavioral stability by centralizing reaction control. It also provides the capability for handling multiple reactions. Reactions terminate when the conditions that invoked them no longer exist. In addition, the commander can override a reaction at any time.

Immediate intervention commands allow the commander to modify the preplanned mission. Immediate intervention commands are similar to fragmentary orders (FRAGOs) used by the Army. The ModSAF commander can either modify the parameters of the current mission or interrupt the current mission to perform a task and then return to the original mission.

An immediate intervention command can be issued through the use of a direct manipulation interface. This interface is provided by allowing the user to modify graphical control measures that are computed by the simulation and then displayed on the map. For example, vehicle position and orientation when occupying a position is computed by the simulation from parameters to the Occupy Position task. The results of this computation are graphically presented by a directed point object on the map display. If the commander prefers a different result, he can use the mouse to drag the point to a new location. He can also change the point's orientation. The vehicle automatically moves to the new< point and take up the new orientation. Similarly, other destination points and routes can be moved.

A.2 Near-Term Movement

ModSAF's approach to near-term movement control, including obstacle avoidance, station keeping, road following, and bridge crossing, is based on an efficient short-term planner. This approach for near-term navigation is based upon the following observations:

- There are many different forms in which movement goals can be expressed, and these forms are not interchangeable. These include cross-country route following and road following, each either individually, or keeping within a formation as part of a group.
- Successful near-term planning requires reasonably stable near-term goals. This means that the form in which the above types of goals are expressed should not require that the goals be frequently updated.
- Obstacle avoidance (including both moving and stationary obstacles) is achieved as a function of the movement goals. In general, there are a large number of courses that can achieve a goal (when you include the ability to turn and change speed). It makes much more sense to use obstacle avoidance considerations in choosing a course from the set of acceptable courses, than to choose an arbitrary course from this set and then modify it to avoid obstacles (which would not necessarily result in a course that can achieve the goal).

- The physical capabilities of a vehicle (such as maximum turn rates, maximum acceleration and deceleration rates, maximum speed, and minimum turn radius) must be considered in all navigation decisions. Choosing courses which cannot be achieved does not yield realistic or representative behavior.
- Sometimes the stated goals cannot be achieved. In such cases, the software that generated the goals must choose new goals. This, in turn, requires that the near-term navigation software be able to detect when a goal cannot be met so that replanning is triggered.
- The fundamental problem with micro-navigation is the combination of multiple constraints. Algorithms that work well for single-constraint cases (avoiding a building or crossing a bridge) fail when supplied with multiple simultaneous constraints (avoiding a building and a tank, or crossing a bridge that is congested with traffic).

The ModSAF system approaches the solution to this difficult problem by simultaneously applying analytic calculus and traditional AI search techniques. All the obstacles (small and fixed, polygonal and fixed, large fixed boundaries, and small moving obstacles) and one goal are placed in a three dimensional map (two spatial dimensions, and one temporal). Short-term plans tracing feasible courses through this map are generated. The planner considers vehicle dynamics to make sure that these courses are physically achievable. These plans are then passed to the dynamics simulation for execution. When the vehicle deviates from the plan significantly or the external environment changes significantly, the planner modifies the plan to take the new conditions into account. Generating realizable courses is essential to minimizing replanning.

For descriptions of the significant calculus and analytic geometry results, the search algorithms, and case studies of the implementation, refer to the paper, "Near-term Movement in ModSAF," by Joshua E. Smith.

A.3 AMSAA Combat Models

ModSAF requires realistic, validated models to adequately support research, training, and combat development. Under the sponsorship of the Simulation Training and Instrumentation Command (STRICOM), certain simulation modules in ModSAF have been enhanced to use the following validated models provided by the Army Materiel Systems Analysis Activity (AMSAA):

1. Target Acquisition
2. Direct Fire Delivery Accuracy
3. Direct Fire Rate of Fire
4. Direct Fire Vulnerability

5. Indirect Fire Vulnerability

The Target Acquisition model determines the performance of individual optical and thermal sensors. It uses the Night Vision Electronics Sensors Directorate (NVESD) methodology and supports four acquisition levels, including detection, classification, recognition, and identification.

The Direct Fire Delivery Accuracy model determines where individual direct fire rounds land, and hence the probability of hit. It uses bias and dispersion inputs to determine the actual impact location for each shot. This is the same approach used by the manned M1A2 simulator.

The Direct Fire Rate of Fire model determines the time between successive shots. It models loading, slewing, ranging, and laying of the gun.

The Direct Fire Vulnerability model determines the damage incurred from a direct fire impact. It is based on the standard kill levels including K, MF, F, and M, and it takes into account the weapon type, impact location and impact orientation.

The Indirect Fire Vulnerability model determines the damage incurred from a nearby indirect fire impact. It can handle high explosive (HE) and Improved Conventional Munition (ICM) effects.

AMSAA has provided standard data structures and algorithms for each of these models, and these have been implemented and documented as part of ModSAF.

The paper, "The Incorporation of Validated AMSAA Combat Models into ModSAF," by Anthony J. Courtemanche and Paul Monday, supplies results for each model on the suitability of the model, implementation difficulties discovered in using the model, and issues in the Verification, Validation and Analysis (VV&A) of the model. Also included are the data requirements needed to support some of the models, and recommendations for DIS protocol modifications to more effectively provide this data.

A.4 Cover and Conceal Algorithms

The need to find covered or concealed positions with respect to enemy locations arises frequently in ModSAF. Examples of tactics that require this ability include occupying a battle position and performing a bounding overwatch maneuver. The ModSAF system is therefore equipped with a cover-finding algorithm and a concealment-finding algorithm.

The cover-finding algorithm takes a rectangular search area and an enemy location as inputs and produces an array of hull-defilade positions as an output. These hull-defilade positions are such that the hull of the vehicle is protected from direct fire by the earth, while leaving the turret exposed. Therefore, the vehicles are able to perform targeting operations and use their firepower from the hull-defilade positions. Intervisibility routines are used to ensure that the enemy location is visible from each covered position.

The concealment-finding algorithm seeks concealment from an enemy location. Like the cover-finding algorithm, the concealment-finding algorithm takes a rectangular search area and an enemy location as inputs and produces an array of points as an output. Although a vehicle in a concealed position is invisible to the enemy, it is not protected from the enemy's direct fire. Therefore, the search for concealment is usually performed only if cover is not found in a given area. This is often the case in flat-terrain environments. The algorithm searches for concealment behind treelines and buildings. If concealment is found behind a treeline, the vehicle can extend its turret through the treeline and use its firepower (intervisibility checks are made from these points to ensure enemy visibility). Concealment behind buildings is used as a last resort, since it is difficult for a vehicle to use its firepower from behind a building.

Both algorithms use the CTDB (compact terrain database) representation of terrain to obtain slope and terrain feature information. The compact representation is favored because more terrain can be stored in physical random access memory, thus decreasing the access time for terrain data. Both algorithms also use a non-preemptive asynchronous ring-based scheduler so that searches are distributed over time while giving the rest of the simulation a chance to run.

A more detailed description of both algorithms can be found in the paper, "Cover and Concealment in ModSAF" by Michael Longtin.

A.5 Benchmarking ModSAF

ModSAF simulation uses a variable time tick that periodically updates each simulated entity. A subjective criteria for the maximum number of entities a SAFsim can handle is defined as the loading at which 90% of the ticks between entity updates occur at intervals of less than .5 second. As the simulation load increases, the update frequency decreases and performance can gradually degrade.

Many factors determine the number of entities that can be simulated by a SAFsim. These include the ModSAF configuration, the number of remote entities being generated by other sources, the types of entities being simulated, the missions that the entities are executing, the terrain the entities

are operating in, the density of the entities on the terrain, and the computer and operating system configurations being used.

ModSAF generates two types of network traffic: simulation packets and PO packets. The level of network traffic that ModSAF generates is a function of computer loading, the entity models used, and their missions.

The paper, "Benchmarking and Optimization of ModSAF 1.0" by Wendy Richardson and G. Robert Vrablik, discusses the number of entities that ModSAF can generate, and the number of network packets generated by and received by ModSAF. This paper references ModSAF 1.0 benchmarking results and indicates expected results for ModSAF 1.2. These benchmarking tests were performed varying the ModSAF configurations, the number of remote entities, and the computers being used. Configurations include the number of ModSAF computers used, their roles, and the network configuration of ModSAF.

A.6 ModSAF 1.5.1 Functionality Matrix

The ModSAF V1.5.1 Functionality Matrix follows:

```
          --Armor/Mechanized Force Tactics and Behaviors--

   Move
      Traveling Overwatch
      Bounding Overwatch
   March
      Pre-Battle
      Road March
   Obstacle awareness
      Terrain
      Vehicle
   Assessment
      Spotter
      Search
      Enemy
   Occupy
      Prep Occupy Position
      Occupy Assembly Area
      Occupy Battle Position
      Hasty Occupy Battle Position
   Defend Position
   Halt in Coil formation
   Targeting
```

```
Formation Keeping
   Column
   Stagger Column
   Echelon Right
   Echelon Left
   Line
   Wedge
   Vee
Follow Unit
Attack
   Assault
   Assault with speed
   React to Contact
   Contact Drill
   Hasty Attack/Action Drill
   Attack By Fire
Targetting Coordination
Fire Control
   Assign priority to different target classes
   Stop Vehicle during fire
   Coordinated fire
   Volley fire
   M-2 raise launcher on halt
Fire Support
Withdrawal using Overwatch
Withdrawal from Minefields
React
   Air Attack
   Artillery
   Contact
Doctrinally correct turret scanning


         --Rotary Wing Aircraft Tactics and Behaviors--

Take-off
Land
Orient Weapons System
Collision Avoidance
Bingo Fuel Return to Base
A/C Resupply At FARP
Hold
   Orbit
   Racetrack
   Hover
Fly Route
   Formation Keeping
Return to Base
Evade and Jink
Level Flight
```

```
Contour Flight
Nap-of-earth Flight
Target Coordination
Occupy Assembly Area
Occupy/Defend Battle Position
RWA Bounding Overwatch
RWA Ground Attack
   Hover Attack
   Running Attack
React to:
   FWA Attack
   RWA Attack
   ADA
   Surface to Air Missile
   Remote Laser Designation


         --Combat Service Support Tactics and Behaviors--

Receive Fuel
Receive Ammunition
Provide Ammunition
Provide Fuel
Tailgate mission
Service Station mission
Cross-leveling
Towing
Repair


         --Air Defense Artillery Tactics and Behaviors--

Track Aircraft
Fire at Aircraft
Follow any ground unit
Occupy Battle Position
Ground Based Sensor provides AD early warning


         --Dismounted Infantry Tactics and Behaviors--

Follow ground unit


         --Combat Engineer Tactics and Behaviors--

Breach Minefields


         --Fire Support Tactics and Behaviors--
```

```
Indirect Fire Missions
   Howitzer
   Mortar
Fire Direction Center
Operator assignment & control


         --Fixed Wing Aircraft Tactics and Behaviors--

Take-off
Land
Orient Weapons System
   Air to air
   Air to Ground
Collision Avoidance
Bingo Fuel RTB
A/C Refuel Rearm
Orbit- Hold at End of Sweep
Racetrack - Hold
Beyond Visual Range Air-to-Air Attack
Combat Air Patrol
Intercept
Lag-Pursuit
Sweep
Egress
Return to Base
Evade and Jink
Level Flight
Low-Level Terrain Flight
Formation Keeping
Target Coordination
FWA Ground Attack
   Direct Attack
   Trailing Attack
   Split up
   Ninety/ten Attack
   Strafe
   Level Attack
   Pop-up Attack
   Lay down
   High Angle Dive Attack
   Medium Angle attack dive
   Low Angle altitude dive
Air-to-Ground Missile
Guns
Rockets
Bombs
Air-to-Air Missiles
```

```
        --User Interface--

Status Change Packets
Execution Matrix
   Saved w/ overlays
   Retrieved
   Edited
   Company/platoon tasks combined
Reports
   Status
   Spot
   Shell
   Reached Control Measure
Minefields
   Placed by Battlemaster
   Artillery delivered
   Environment Map Queries


        --Additional Architecture Capabilities--

Multi-Level Terrain
   Drive over elevated bridges
Radio Communications


               --ModSAF Objects--

Armored Vehicles
   M-1
   M1A1
   M1A2
   T-72
   T-80
   BTR-80
   BRDM-2
   LUCHS
   LEO-1A5
   LEO-2
   MARDER-1A3
   3 LEO-1A5 /1 MARDER Mixed Plt
   3 LEO-2 /1 MARDER Mixed Plt
   1 LEO-2 /1 MARDER Mixed Plt
   JAGUAR-1
   BRDM-2/BTR-80 Mixed Plt
   BMP-1/BRDM-2 Mixed Plt
```

```
    BMP-2/BRDM-2 Mixed Plt
    T-80/BRDM-2 Mixed Plt
    BTR60

Mechanized Vehicles
    M-2
    NLOS
    LOSAT
    M-113 APC
    M-113 SKORPION Eng. Vehicle
    M-113 Observer
    M-113 Ambulance
    M-977 HEMTT Ammo Carrier
    M-978 HEMTT Fuel Tanker
    M577 HMMWV



    M88 Recovery Vehicle
    BMP-1
    BMP-2
    BTR-60
    Red Motorized Rifle
    Reinforced
    Blue Mechanized Infantry
    Attached Tank Plt

Rotary Wing Aircraft
    AH-1
    AH-64
    RAH-66 Comanche
    OH-58 Kiowa
    Mi-24 Hind
    Mi-28 Hokem
    Mi-8 Havoc

Fixed Wing Aircraft
    F-14
    F-16
    A-10
    MIG-27
    MIG-29
    SU-25

Air Defense Artillery
    ZSU-23/4
    SA-9 w/ tracked chassis
    M-2 Stinger
    Avenger
    GBS
```

```
Fire Support
   Russian 1V13
   Russian 1V14
   Russian 1V15
   Russian 1V16
   Russian 2B11
   Russian 2S1
   Russian 2S19
   Russian 2S6
   Russian ACRV1V16
   Russian URAL 375C
   Russian URAL 375F
   US M109
   US M109A3
   US M109A5
   US M109A6
   US M270
   US M270GAT2
   US M270M26
   US M270M77
   M-106A1 Mortar
   MLRS
   Counter Battery Radio

Dismounted Infantry
   Russian - DI Squad
   Russian - DI - MG Squad
   Russian - DI - Lfrk-5 Squad
   Russian - DI - Ags-17 Squad
   FGR - DI - MG Squad
   FGR - DI - Milan Squad
   BMP-1 DI Platoon
   BMP-2 DI Platoon
   BTR-80 DI Platoon
   US Rifle Squad

Missiles
   US Phoenix
   US Sidewinder
   US Sparrow
   US Stinger
   US TOW
   US Hellfire
   US Maverick
   US Javelin
   US Dragon
   German Milan
   Russian Sagger
   Russian Songster
```

```
    Russian Spandrel
    Russian Spigot
    Russian Gaskin
    Russian Hot
    Russian Alamo
    Russian Archer
    Russian SA15
    Russian SA9

 Munitions
    German AT2
    German 120HEAT
    German 120SABOT
    German 35AP
    German 35HEI
    German DM63
    German DM81
    German Panzerfaust
    Russian 125HEAT
    Russian 125SABOT
    Russian 127AA
    Russian 127MG
    Russian 145MG
    Russian 23AP
    Russian 30HE
    Russian 3023
    Russian 73HEAT
    Russian 9K25
    Russian BK 6M
    Russian BK M
    Russian D
    Russian D 462
    Russian D 540
    Russian Gaskin
    Russian M 21 OF
    Russian OF843B
    Russian OF 45
    Russian OF 462
    Russian OF 540
    Russian OF 61
    Russian OF 843B
    Russian PS
    Russian ROCKET 64
    Russian S5
    Russian VOG 17M
    US Hydra70 M151
    US Hydra70 M261
    US L8A1
    US M107
    US M110E2
```

```
US M26
US M329A1
US M329A2
US M392A2
US M449A1
US M456A1
US M483A1
US M50
US M549A1
US M59
US M692
US M712
US M718
US M731
US M741
US M76
US M77
US M789
US M792
US M8
US M855
US MX943
US Mk82
US STAFF
US Sadarm
```

Index

N

O

P

Q

R

S

T

U

V

W

Y

Z